Control-based Finite-element Model Updating of Structures

by

Paul Paquet

A thesis

presented to the University of Waterloo

in fulfilment of the

thesis requirement for the degree of

Master of Applied Science

in

Civil Engineering

Waterloo, Ontario, Canada, 2009

Declaration

I hereby declare that I am the sole author of this thesis. This is a true copy of the thesis, including any required final revisions, as accepted by my examiners.

I understand that my thesis may be made electronically available to the public.

Abstract

Finite-element model updating is the process of using measured data from a structure to update a numerical model representation of the structure. The measured data can represent either the static or dynamic properties of the structure. This document reviews and evaluates several methods of finite-element (FE) model updating, including direct, indirect, and control-based methods for the dynamic case. It is important to have a correct finite-element model obtained using model updating methods either to assess the current condition, or to modify the structure from its current state.

In this study, three types of methods were evaluated; direct, indirect, and control based finite-element model updating methods. Each method was first used to update a simple example model for two separate cases. For the first case, the entire set of measured modal parameters were used; and for the second case, only a sub-set of the eigenvalues were used. These examples provide insights into the advantages and disadvantages of various methods.

The model updating methods are also used to update a full-scale 42 degree of freedom model. Since it is not practical to measure all the degrees of freedom, the model was reduced using the SEREP model reduction method, down to 18 degrees of freedom. This was done to evaluate the effectiveness of the model updating methods on a real structure. Detailed methodologies and a comparison between the relative advantages and disadvantages between various model updating methods are highlighted in this thesis.

Acknowledgements

First, I would like to thank Dr. Sriram Narasimhan for his expertise in the subject and help in completing this research. Secondly, I would like to thank the Structural Dynamics and Control group; Mr. Aaron Roffel, for his work modelling the control tower and his aid with MATLAB scripts; and Mr. Budhayita Hazra, for his knowledge in control theory. And lastly, I would like to thank the University of Waterloo Civil and Environmental Engineering Faculty for its support over the last seven years.

I would also like to thank my parents, sister, grandmother, and friends for the years of support they have provided. Without all of you, I would not be where I am today.

I am dedicating this thesis to my grandfather; I wish he were around to witness this accomplishment. Guy Charbonneau, I love you and may you rest in peace.

TABLE OF CONTENTS

1.0 Introduction

The use of Finite-Element Models (FEM) has become common place in structural analysis and design. Commercial FEM programs such as SAP2000© and ANSYS© allow designers to create numerical representations of structures with relative ease. However, these models often do not represent the actual physical characteristics of the structure or the results obtained from experiments. There are many factors that affect the accuracy of the numerical model, these include but are not limited to: simplifying assumptions regarding the boundary conditions, unplanned loads on the structure, and material imperfections. The objective of model updating is to adjust the numerical model of the structure so that the model predictions are in agreement with the test results. Implicit in many model-updating methods is that the test results are in-violate and represent the "true" characteristics of the actual structure.

The most common form of describing the dynamics of a structure is through the use of a second order differential equation that is shown in matrix form in equation (1.1)

$$\mathbf{M}\ddot{\mathbf{x}} + \mathbf{C}\dot{\mathbf{x}} + \mathbf{K}\mathbf{x} = \mathbf{f}(t) \tag{1.1}$$

where	$\mathbf{M}$	is the Finite-element mass matrix,
	$\mathbf{C}$	is the Finite-element damping matrix,
	$\mathbf{K}$	is the Finite-element stiffness matrix,
	$\ddot{\mathbf{x}}$	is a vector of acceleration responses,
	$\dot{\mathbf{x}}$	is a vector of velocity responses,
	$\mathbf{x}$	is a vector of displacement responses, and
	$\mathbf{f}(t)$	is the external disturbance vector.

The mass, damping, and stiffness matrices are created by using the estimated properties of the structure in the equations of motion. Typically, the process of estimating the structure properties starts with creating a finite-element (FE) model with the best structural information available. The model is then reduced to a limited number of degrees of freedom in order reduce the computational burden of the calculations. Formation of the damping matrix, **C**, is often hard, and one has to resort to estimates for the magnitude and structure of the matrix that is based on experience and other acceptable norms. For the present, consider the formation of the mass and stiffness matrices, called **M** and **K** of size *n*. If a discrete representation (or lumped parameter model) of a structure shown in Figure 1 (and also in Figure 2) is considered, the mass and stiffness matrices can be found by the principles of structural analysis as:

$$\mathbf{M} = \begin{bmatrix} m_1 & 0 & 0 & 0 \\ 0 & m_2 & 0 & 0 \\ 0 & 0 & m_3 & 0 \\ 0 & 0 & 0 & m_4 \end{bmatrix}; \ \mathbf{K} = \begin{bmatrix} k_1 + k_2 & -k_2 & 0 & 0 \\ -k_2 & k_2 + k_3 & -k_3 & 0 \\ 0 & -k_3 & k_3 + k_4 & -k_4 \\ 0 & 0 & -k_4 & k_4 \end{bmatrix};$$
$$\mathbf{C} = \begin{bmatrix} c_1 + c_2 & -c_2 & 0 & 0 \\ -c_2 & c_2 + c_3 & -c_3 & 0 \\ 0 & -c_3 & c_3 + c_4 & -c_4 \\ 0 & 0 & -c_4 & c_4 \end{bmatrix} \tag{1.2}$$

For structural systems, **M** is positive definite and **K** is, at least, semi-definite.

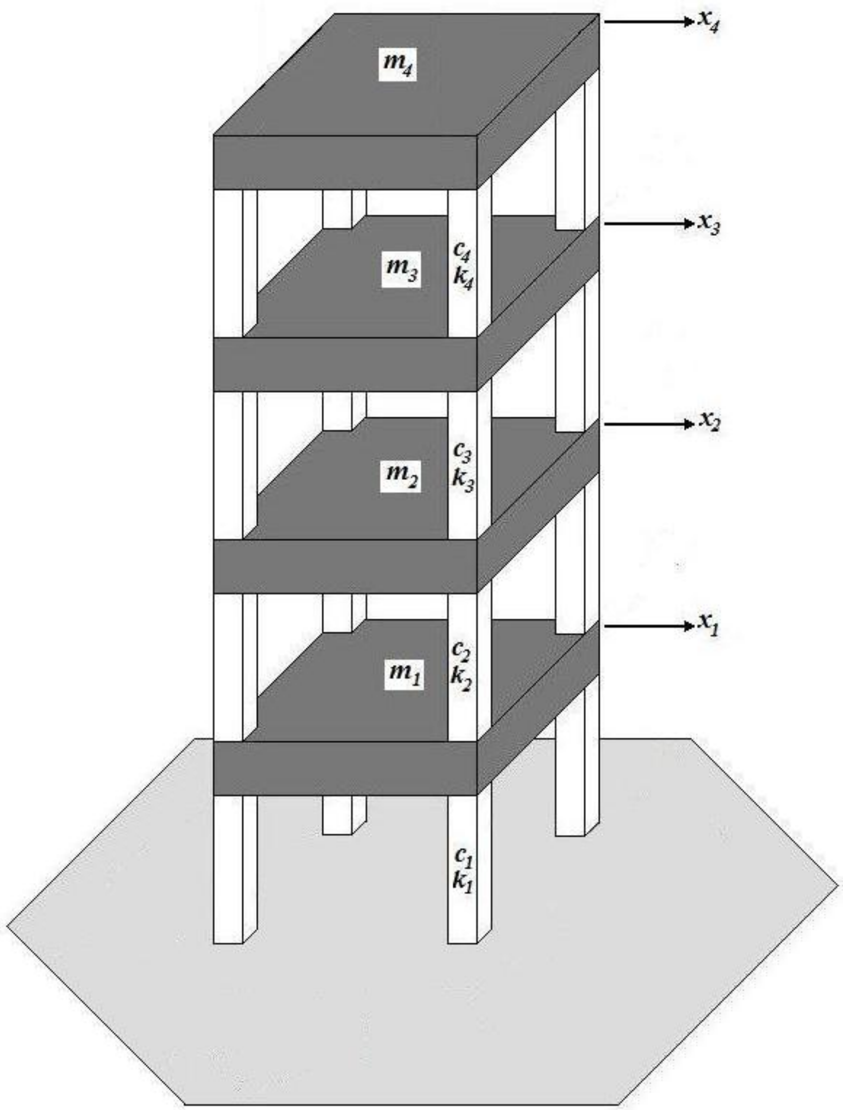

Figure 1: Four-storey Structure Example

Using separation of variables,

$$\mathbf{x}(t) = \boldsymbol{\varphi} e^{\lambda t}, \ \boldsymbol{\varphi} \text{ is a constant vector} \tag{1.3}$$

And substituting in equation (1.1), we get

$$p(\lambda)\boldsymbol{\varphi} = 0 \tag{1.4}$$

where $p(\lambda) = \lambda^2 \mathbf{M} + \lambda\mathbf{C} + \mathbf{K}$ is called the "quadratic pencil". The roots of the polynomial $p(\lambda)$, which are generally complex, are called the eigenvalues and the corresponding $\boldsymbol{\varphi}$ satisfying equation (1.4) are the eigenvectors of (1.4). Equation (1.4) in standard physical co-ordinates is of dimension *n*, often referred to as the standard form. Alternatively, equation (1.4) can also be written in state-space form, where the dimension is *2n*, instead of *n*. That is, there are 2*n* eigenvalues and 2*n* eigenvectors, where *n* is the order of the system, or equivalently the degrees of freedom (DOF) of the FE model.

For the case of an un-damped system, the quadratic pencil reduces to

$$\left[\mathbf{K} - \lambda^2\mathbf{M}\right]\boldsymbol{\varphi} = \mathbf{0}. \tag{1.5}$$

As can be seen from equation (1.5), the eigenvalues and eigenvectors are intricately related to the system properties **K** and **M**. Hence, in traditional modal analysis and model updating, the primary approach is to adjust **K** and **M** to reflect $\boldsymbol{\lambda}$ and $\boldsymbol{\varphi}$ observed using test data. Most of the model updating methods employ a two-step approach. First, **M** and **K** matrices are updated using the measured $\boldsymbol{\lambda}$ and $\boldsymbol{\varphi}$, then, the damping matrix **C** is calculated using **M** and **K**; $\boldsymbol{\lambda}$ and $\boldsymbol{\varphi}$ are considered in-violate. This concept will be discussed in the next chapter.

Model-updating methods can be broadly classified as belonging to one of these categories: (i) constrained optimization matrix update methods, (ii) penalty function matrix update, and (iii) control-based eigen-structure assignment methods. Methods based on (i) and (ii) have been studied extensively in the literature (Friswell, 1995). The main objective of this thesis is to study the eigen-structure assignment methods and their applicability to structural model updating problems. The purpose of the eigen-structure assignment methods is to have the capability of updating only a portion of the structural system, while leaving the rest undisturbed.

The main objectives of this thesis are to (i) apply the eigen-structure assignment methods to the problem of structural model updating and (ii) compare eigen-structure and popular optimization methods of model updating.

The scope of this thesis in limited to updating the structural system matrices to represent the eigen-data obtained from experiments. Though the concepts of model updating could be extended to represent quantities such as modal displacement, velocities, and accelerations, this is not explored in this thesis.

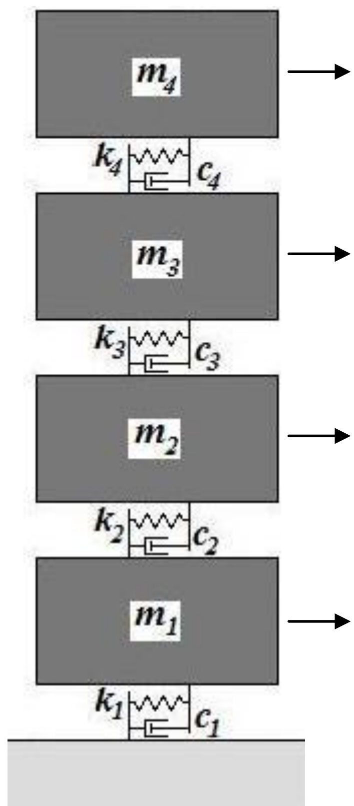

Figure 2: Simplified Four-storey Structural Example

2.0 Background

This chapter begins with a brief summary of model reduction/expansion methods to reconcile the size of mode shapes (eigenvectors) of FE and experimental models. The main methods of model updating that have been studied in the literature are then reviewed. Two main methods: (i) constrained optimization matrix update method and (ii) sensitivity based penalty function methods are reviewed first. A numerical example is also presented to illustrate the steps involved in each of these methods.

2.1 Modal Assurance Criterion

The modal assurance criterion (MAC) is a technique used to evaluate the correlation between mode shapes. In this document it is used to compare the mode shapes of the updated analytical model with the mode shapes from the measured system. The MAC between a measured mode, $\boldsymbol{\varphi}_{mj}$, and an analytical mode, $\boldsymbol{\varphi}_{aj}$, is defined as

$$\mathbf{MAC} = \frac{\left|\boldsymbol{\varphi}_{mj}^{T}\boldsymbol{\varphi}_{aj}\right|^{2}}{\left(\boldsymbol{\varphi}_{aj}^{T}\boldsymbol{\varphi}_{aj}\right)\left(\boldsymbol{\varphi}_{mj}^{T}\boldsymbol{\varphi}_{mj}\right)}. \tag{2.1}$$

The value of the MAC is between zero and one; a value of one means that the analytical mode-shape is a multiple of the measured mode shape. Whereas, a value of zero means that there is no correlation between the two. Traditionally having a MAC value of 0.95 or higher is considered to be acceptable (Friswell, 1995).

2.2 Model Reduction/Expansion

Typically, FE models of structures have several degrees of freedom. It is impractical to measure the responses of every degree of freedom. As well, the dominant modes are usually the first few, and the higher natural frequencies and modes contribute little to the overall dynamic response. Because of these reasons, it is beneficial to reduce the number of degrees of freedom to include only the most important ones. This is referred to as Model Reduction. An analogous situation occurs when the experimentally obtained modes are statically incomplete, and the dimensions of these vectors are to be increased to match the dimensions of the FE model. This is a modal “expansion” process.

Many methods of reducing models exist at this time, and each has advantages and disadvantages associated with it. This section will introduce four of these methods: Static Reduction, the Improved Reduction System, the Iterated Improved Reduction System, and the System Equivalent Reduction Expansion Process (which can be used for modal expansion as well).

The methods described in this section separate and re-order the original mass and stiffness matrices into master and slave co-ordinates. Master co-ordinates are those degrees of freedom that affect the mode shapes the most. The master co-ordinates are chosen where the inertia effects are high and the stiffness is low, whereas, the slave co-ordinates are chosen where the inertia is low and the stiffness is high. This is because it is assumed that the inertia forces of the slave co-ordinates are negligible compared to their elastic forces. Identifying the master and slave can be completed by observing the ratio of the diagonal terms in the mass and stiffness matrices, k_{ii}/m_{ii} (Friswell, 1997). The master co-ordinates will be the co-ordinates that are retained in the reduced mass and stiffness matrices, and the slave co-ordinates, which affect the mode shapes the least, will be removed from the system.

Early model reduction methods are based on static condensation of the FE degrees of freedom. One such standard method is Guyan reduction, (Guyan, 1965) which will be presented first. Following this, more powerful dynamic condensation methods, such as the Improved Reduction System method, the Iterated Improved Reduction System, and the System Equivalent Reduction Expansion Process will be presented. For each of the model reduction methods, the system is assumed to be un-damped.

2.2.1 Model Reduction Example

In order to illustrate the workings of the model reduction/expansion methods, a common numerical test bed is used. Model reduction methods will be used to reduce the vertical and torsional stiffness and mass components from the following example of a four-storey shear-beam model. In this case, each storey is assumed to have two degrees of freedom. The stiffness and mass matrices for the analytical model are as follows:

$$\mathbf{K} = \begin{bmatrix} 10790.9 & -5395 & 0 & 0 & 0 & 0 & 0 & 0 \\ -5395 & 10790.9 & 0 & 0 & -5395 & 0 & 0 & 0 \\ 0 & 0 & 200000 & 0 & 0 & 0 & -100000 & 0 \\ 0 & 0 & 0 & 7241.6 & 0 & 0 & 0 & -3620.8 \\ 0 & -5395 & 0 & 0 & 10790.9 & -5395 & 0 & 0 \\ 0 & 0 & 0 & 0 & -5395 & 5395 & 0 & 0 \\ 0 & 0 & -100000 & 0 & 0 & 0 & 100000 & 0 \\ 0 & 0 & 0 & -3620.9 & 0 & 0 & 0 & 3620.8 \end{bmatrix} \tag{2.2}$$

$$
\mathbf{M} = \begin{bmatrix} 60 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 60 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 60 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0.929 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 60 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 60 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 60 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0.929 \end{bmatrix} \tag{2.3}
$$

The resulting eigenvalues and eigenvectors, by solving the eigenvalue problem posed earlier (1.4), are as follows:

$$
\mathbf{L} = \begin{bmatrix} 10.85 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 89.92 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 211.08 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 317.62 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 636.61 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1488.67 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 4363.4 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 10203.5 \end{bmatrix} \tag{2.4}
$$

$$
\mathbf{V} = \begin{bmatrix} -0.029 & 0.075 & 0.085 & -0.055 & 0 & 0 & 0 & 0 \\ -0.055 & 0.075 & -0.029 & 0.085 & 0 & 0 & 0 & 0 \\ -0.075 & 0 & -0.075 & -0.075 & 0 & 0 & 0 & 0 \\ -0.085 & -0.075 & 0.055 & 0.029 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0.068 & 0 & 0.11 & 0 \\ 0 & 0 & 0 & 0 & 0 & -0.55 & 0 & -0.88 \\ 0 & 0 & 0 & 0 & 0.11 & 0 & -0.068 & 0 \\ 0 & 0 & 0 & 0 & 0 & -0.88 & 0 & 0.55 \end{bmatrix} \tag{2.5}
$$

Where **L** represents the diagonal matrix of eigenvalues, λ, and **V** represents the matrix of eigenvectors, $\boldsymbol{\varphi}$. The results of each model reduction method will be discussed in section 2.2.6.

2.2.2 Guyan or Static Reduction

One of the simplest reduction methods is static reduction. It was introduced by Guyan in 1965 (Guyan, 1965). The method assumes that no force is applied at the slave degrees of freedom and that the system is un-damped. As previous stated, the mass and stiffness matrices are separated into master and slave co-ordinates, and the equation of motions becomes

$$\begin{bmatrix} \mathbf{M}_{mm} & \mathbf{M}_{ms} \\ \mathbf{M}_{sm} & \mathbf{M}_{ss} \end{bmatrix} \begin{Bmatrix} \ddot{\mathbf{x}}_m \\ \ddot{\mathbf{x}}_s \end{Bmatrix} + \begin{bmatrix} \mathbf{K}_{mm} & \mathbf{K}_{ms} \\ \mathbf{K}_{sm} & \mathbf{K}_{ss} \end{bmatrix} \begin{Bmatrix} \mathbf{x}_m \\ \mathbf{x}_s \end{Bmatrix} = \begin{Bmatrix} \mathbf{f}_m \\ \mathbf{0} \end{Bmatrix} \tag{2.6}$$

Where, the subscripts *m* and *s* refer to the master and slave co-ordinates, respectively. The method then ignores the inertia terms in the second set of equations (hence, the word “static” is used), resulting in

$$\mathbf{K}_{sm}\,\mathbf{x}_m + \mathbf{K}_{ss}\,\mathbf{x}_s = \mathbf{0} \tag{2.7}$$

By eliminating the slave degrees of freedom, we obtain:

$$\begin{Bmatrix} \mathbf{x}_m \\ \mathbf{x}_s \end{Bmatrix} = \begin{bmatrix} \mathbf{I} \\ -\mathbf{K}_{ss}^{-1} \cdot \mathbf{K}_{sm} \end{bmatrix} \{\mathbf{x}_m\} = \mathbf{T}_s\,\mathbf{x}_m \tag{2.8}$$

where $\mathbf{T}_s$ is the static transformation matrix. The transformation matrix has the same number of rows as the original mass and stiffness matrices, however the number of columns are the same as the order to which the system is being reduced to. The reduced mass and stiffness matrices are obtained using:

$$\mathbf{M}_R = \mathbf{T}_S^{\mathbf{T}}\,\mathbf{M}\,\mathbf{T}_S \qquad\qquad \mathbf{K}_R = \mathbf{T}_S^{\mathbf{T}}\,\mathbf{K}\,\mathbf{T}_S \tag{2.9}$$

These reduced mass and stiffness matrices will produce similar eigenvalues for lower modes: for the higher modes, however, the error increases for higher frequencies. For simple buildings (having a small number of degrees of freedom), this method provides satisfactory results. For larger buildings, where it is necessary to reduce many slave degrees of freedom, this method will not be as accurate as some of the more advanced methods presented next.

2.2.3 Improved Reduction System

A method known as the Improved Reduction System (IRS) was introduced by O'Callahan in 1989 (Friswell, 1995). This method is an improvement over the Guyan static reduction method by introducing a term that includes the inertial effects as pseudo static forces. A transformation matrix, $\mathbf{T}_i$, is used to reduce the mass and stiffness matrices. It is defined as

$$\mathbf{T}_i = \mathbf{T}_S + \mathbf{S}\,\mathbf{M}\,\mathbf{T}_S\,\mathbf{M}_R^{-1}\,\mathbf{K}_R \tag{2.10}$$

where

$$\mathbf{S} = \begin{bmatrix} \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{K}_{SS}^{-1} \end{bmatrix} \tag{2.11}$$

and $\mathbf{M}_{\boldsymbol{R}}$ and $\mathbf{K}_{\boldsymbol{R}}$ are the statically reduced mass and stiffness matrices.

The new reduced mass and stiffness matrices can be found by

$$\mathbf{M}_{IRS} = \mathbf{T}_i^{\mathbf{T}}\,\mathbf{M}\,\mathbf{T}_i \qquad\qquad \mathbf{K}_{IRS} = \mathbf{T}_i^{\mathbf{T}}\,\mathbf{K}\,\mathbf{T}_i \tag{2.12}$$

where **M** and **K** are the original mass and stiffness matrices.

For this method, the rows and columns corresponding to the slave co-ordinates are removed from the mass and stiffness matrices one at a time; this allows the mass and stiffness matrices to adapt to the removal of a slave, and can possibly change the degree of freedom that will be removed. After each reduction, the degree of freedom with the lowest k_{ii}/m_{ii} term is the slave which will be removed next.

2.2.4 Iterated Improved Reduction System

Friswell (1998) presented an iterative IRS method which improves the matrix reductions by ensuring that the transformation matrix for each reduction is optimized. This method is very similar to the IRS, the difference is that the transformation matrix, $\mathbf{T}_{\boldsymbol{i}}$, for each slave co-ordinate being reduced is found iteratively until it converges, this is advantageous in larger systems since there may be several correct transformation matrices, but only one optimal. The equation to find $\mathbf{T}_{\boldsymbol{i}}$ is similar to equation (2.10), the only change is the $\mathbf{T}_{\boldsymbol{s}}$ in the second term of the IRS method is now a $\mathbf{T}_{\boldsymbol{i}}$ and the reduced mass and stiffness matrices are updated on each iteration.

$$\mathbf{T}_{i+1} = \mathbf{T}_S + \mathbf{S}\,\mathbf{M}\,\mathbf{T}_i\,\mathbf{M}_{Ri}^{-1}\,\mathbf{K}_{Ri} \tag{2.13}$$

These iterations continue until $\mathbf{T}_{\boldsymbol{i+1}} = \mathbf{T}_{\boldsymbol{i}}$. That value of $\mathbf{T}$ is then used as the final transformation matrix for that reduction.

$$\mathbf{M}_R = \mathbf{T}^{\mathrm{T}}\,\mathbf{M}\,\mathbf{T} \qquad\qquad \mathbf{K}_R = \mathbf{T}^{\mathrm{T}}\,\mathbf{K}\,\mathbf{T} \tag{2.14}$$

2.2.5 System Equivalent Reduction Expansion Process

O'Callahan et al. (1989) introduced the System Equivalent Reduction Expansion Process (SEREP) model reduction/expansion technique.

An un-damped system with the displacement defined as

$$\mathbf{x} = \mathbf{\Phi q} \tag{2.15}$$

where $\mathbf{\Phi}$ is defined as the matrix containing the eigenvectors and can be organized such that the master co-ordinates, denoted by '*m*', are placed in the upper part of the vector and the slave co-ordinates, denoted by '*s*', are placed in the lower section of the vector as such

$$\begin{Bmatrix} \mathbf{x}_m \\ \mathbf{x}_s \end{Bmatrix} = \begin{Bmatrix} \mathbf{\Phi}_m \\ \mathbf{\Phi}_s \end{Bmatrix} \mathbf{q} \quad (2.16)$$

It can be shown that the pseudo-inverse of the modal matrix containing the master co-ordinates is

$$\mathbf{\Phi}_m^+ = \left(\mathbf{\Phi}_m^{\mathbf{T}} \cdot \mathbf{\Phi}_m \right)^{-1} \mathbf{\Phi}_m^{\mathbf{T}} . \quad (2.17)$$

Considering only the master co-ordinates, the modal displacement can be solved using equation (2.18)

$$\mathbf{q} = \mathbf{\Phi}_m^{\mathbf{T}} \, \mathbf{x}_m . \quad (2.18)$$

Substituting equation (2.18) into (2.16) produces an expression, equation (2.19), for the full system’s displacement vector in terms of the reduced system’s displacement vector

$$\mathbf{x} = \begin{Bmatrix} \mathbf{\Phi}_m \\ \mathbf{\Phi}_s \end{Bmatrix} \mathbf{\Phi}_m^{\mathbf{T}} \, \mathbf{x}_m . \quad (2.19)$$

So, the global transformation matrix is then defined as

$$\mathbf{T}_U = \begin{bmatrix} \mathbf{\Phi}_m \\ \mathbf{\Phi}_s \end{bmatrix} \mathbf{\Phi}_m^+ \quad (2.20)$$

The reduced mass and stiffness matrices are then calculated using equations (2.21) and (2.22) respectively.

$$\mathbf{M}_R = \mathbf{T}_U^{\mathbf{T}} \mathbf{M} \mathbf{T}_U \quad (2.21)$$

$$\mathbf{K}_R = \mathbf{T}_U^{\mathbf{T}} \mathbf{K} \mathbf{T}_U \quad (2.22)$$

Since a single transformation matrix is used, this process is also reversible; using equation (2.23) will reverse the model reduction, and is the basis of model expansion.

$$\begin{Bmatrix} \mathbf{\Phi}_m \\ \mathbf{\Phi}_s \end{Bmatrix} = \mathbf{T}_U \mathbf{\Phi}_m^+ \tag{2.23}$$

It has been shown by Friswell (1998) that the transformation matrix found using the Iterated IRS method will converge to the transformation matrix calculated in the SEREP method. The MATLAB script for the SEREP model reduction can be found in Appendix A4.

2.2.6 Results

The results for the model reduction techniques are achieved by using the reduced system matrices to find the frequency response of the system. Since this example is small, it is not unexpected that all the methods produce the same stiffness and mass matrices.

Figure 3 displays the frequency response of the reduced and original systems. The four smallest eigenvalues are approximately the same for both systems, but the amplitudes are different. The original system does show small peaks in amplitude for the larger eigenvalues, but it can be seen that they are not nearly as influential as the first four eigenvalues. The fact that the amplitudes of the last four eigenvalues are nearly zero is a good example of why model reduction can be effective, since the larger eigenvalues are often not as important to a system.

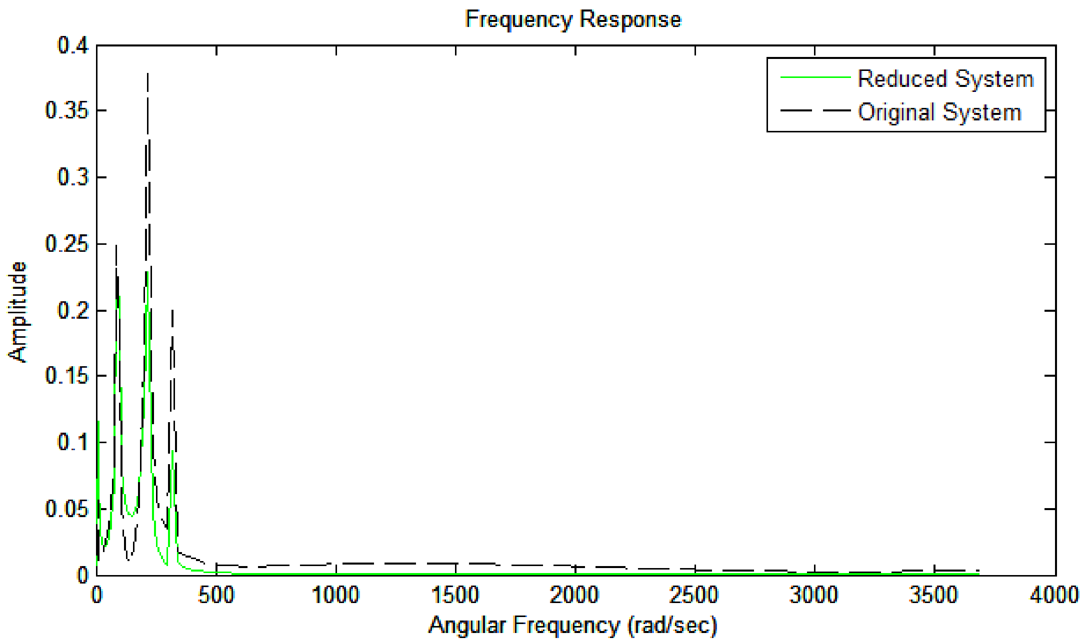

Figure 3: Frequency Response

In order to observe the differences in amplitudes for the first four eigenvalues, Figure 4 displays the same graph, but focuses on the frequency responses up to 400 rad/sec. The response for the reduced system contains nearly identical peaks in terms of angular frequency; however, the amplitudes of the first, third, and fourth eigenvalues are considerably different.

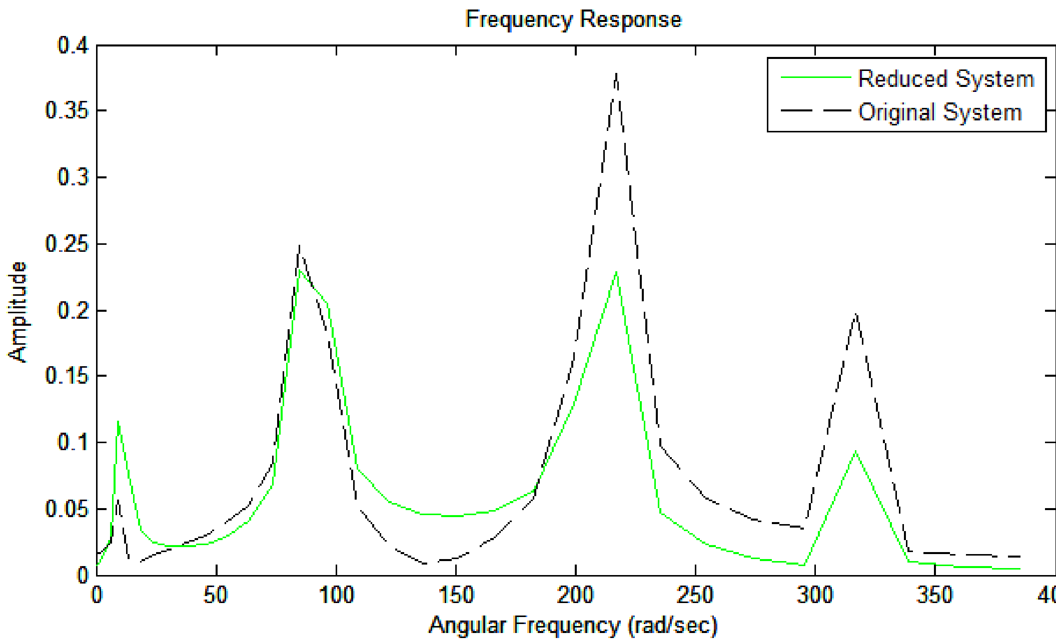

Figure 4: Frequency Response (First Four Eigenvalues)

Despite yielding identical reduced systems for this example, each model reduction method has advantages and disadvantages. When dealing with real structures that may not contain ideal mass and stiffness matrices, only the iterated IRS and SEREP methods are able to re-produce the desired eigenvalues and eigenvectors.

The main advantage that the iterated IRS method has is the ability to re-evaluate which degree of freedom is the least relevant to the system. After each iteration, the re-distribution of mass and stiffness matrices may depend more on the specific degree of freedom. This method can also be used to determine the optimal location and direction to place transducers.

The advantage of the SEREP method is that it is computationally easier, and if the slave degrees of freedom are already chosen, will produce the same result as the iterated IRS method.

The reduced mass and stiffness matrices for the four-storey system are:

$$
\mathbf{K}_R = \begin{bmatrix} 10790.92 & -5395.46 & 0 & 0 \\ -5395.46 & 10790.92 & -5395.46 & 0 \\ 0 & -5395.46 & 10790.92 & -5395.46 \\ 0 & 0 & -5395.46 & 5395.46 \end{bmatrix} \tag{2.24}
$$

$$
\mathbf{M}_R = \begin{bmatrix} 60 & 0 & 0 & 0 \\ 0 & 60 & 0 & 0 \\ 0 & 0 & 60 & 0 \\ 0 & 0 & 0 & 60 \end{bmatrix} \tag{2.25}
$$

2.3 Lagrange Multiplier Methods

This section begins with a literature review of some of the important works in the areas of Lagrange multiplier based matrix updating methods and the penalty function based methods. It should be noted that in optimization, Lagrange multiplier methods involve strict imposition of constraints, whereas the penalty function methods allow flexibility to the user, or "self" constraints. A common numerical example is presented to illustrate the steps involved in both these methods.

2.3.1 Four-Storey Example

The four degree of freedom system's mass and stiffness matrices are those found in equations (2.24) and (2.25).

Each of the methods presented in Chapter 2 were applied to the example model twice; the first application included all of the measured eigenvalues and eigenvectors. However, the second application simulated having transducers on the first and third degrees of freedom only. Only the first two measured eigenvalues were used, and the first and third degrees of freedom of the two corresponding eigenvectors were given.

The eigenvalues, **L**, and eigenvectors, **V**, of the system were calculated using the eigenvalue problem equations (1.4) and (1.5) to be

$$\mathbf{L} = \begin{bmatrix} 10.85 & 0 & 0 & 0 \\ 0 & 89.92 & 0 & 0 \\ 0 & 0 & 211.08 & 0 \\ 0 & 0 & 0 & 317.62 \end{bmatrix} \tag{2.26}$$

$$\mathbf{V} = \begin{bmatrix} -0.029 & 0.075 & 0.085 & -0.055 \\ -0.055 & 0.075 & -0.029 & 0.085 \\ -0.075 & 0 & -0.075 & -0.075 \\ -0.085 & -0.075 & 0.055 & 0.029 \end{bmatrix} \tag{2.27}$$

To create the simulated eigenvalues and eigenvectors, the stiffness matrix was changed by arbitrarily assigning values as seen in equation (2.28). The resulting eigenvalues and eigenvectors are as follows:

$$\mathbf{K_m} = \begin{bmatrix} 10000 & -5200 & 0 & 0 \\ -5200 & 10000 & -5200 & 0 \\ 0 & -5200 & 10000 & -5200 \\ 0 & 0 & -5200 & 5200 \end{bmatrix} \tag{2.28}$$

$$\mathbf{L_m} = \begin{bmatrix} 6.56 & 0 & 0 & 0 \\ 0 & 82.28 & 0 & 0 \\ 0 & 0 & 198.03 & 0 \\ 0 & 0 & 0 & 299.81 \end{bmatrix} \tag{2.29}$$

$$\mathbf{V_m} = \begin{bmatrix} -0.031 & 0.075 & 0.085 & -0.055 \\ -0.058 & 0.073 & -0.031 & 0.085 \\ -0.075 & -0.004 & -0.073 & -0.075 \\ -0.082 & -0.076 & 0.057 & 0.030 \end{bmatrix} \tag{2.30}$$

The example was chosen to evaluate each method for a basic structural setting. However, the second application was chosen to evaluate the ability of each method when only a portion of the modal data is available. Figure 5 visually represents the second application. The second and fourth storeys were outfitted with accelerometers. Because of this, it is only possible to measure two eigenvalues and only the eigenvector values representing those two storeys.

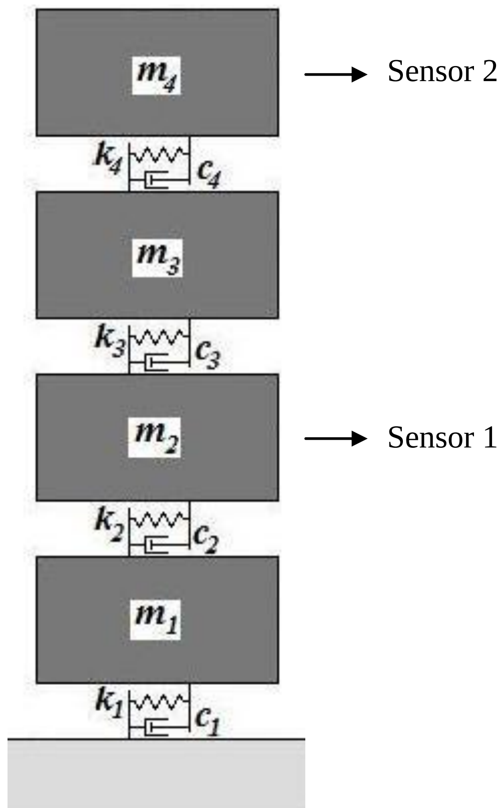

Figure 5: Simplified Model With Two DOFs Measured

The measured eigenvalues and eigenvectors for this application are presented in equations (2.31) and (2.32).

$$\mathbf{L_m} = \begin{bmatrix} 6.56 & 0 \\ 0 & 82.28 \end{bmatrix} \tag{2.31}$$

$$\mathbf{V_m} = \begin{bmatrix} -0.031 & 0.075 \\ -0.075 & -0.004 \end{bmatrix} \tag{2.32}$$

The direct model updating methods do not have the ability to update the FEM without complete modal data. In order to provide full modal data, modal expansion was completed on the two measured eigenvectors; the expanded eigenvectors were calculated to be

$$\mathbf{V_m} = \begin{bmatrix} -0.031 & 0.075 \\ -0.058 & 0.073 \\ -0.075 & -0.004 \\ -0.084 & -0.081 \end{bmatrix}. \tag{2.33}$$

However, since all four degrees of freedom are required to be present, the analytical third and fourth eigenvalues and eigenvectors were included in the measured modal data.

The objective of this chapter is to gain a better understanding of each method and to observe the ability of each method to update a system with incomplete modal data.

2.3.2 Lagrange Multiplier

The Lagrange multiplier based methods (also known as reference based methods or direct methods) generally consider one parameter set, either mass or stiffness to be correct, and the remaining two, that is either mass or stiffness, and the modes, are updated by minimizing a cost function with the appropriate constraints imposed through Lagrange multipliers. In the following discussion the key steps are presented along with a numerical example that assume that the mass is correct.

Modes that are measured from the structure will not necessarily be orthogonal to the analytical mass matrix since there are likely fewer transducers than degrees of freedom and because of imperfect measurements. For the direct methods that assume that the mass matrix is correct, it is usually difficult to enforce orthogonality. In order to ensure the eigenvectors are orthogonal, the measured eigenvectors must be corrected. Baruch (1978) has derived a cost function, *J*, (2.34), in which the newly updated eigenvector matrix **Φ** is to be minimized

$$J = \left\| \mathbf{N}(\mathbf{\Phi} - \mathbf{\Phi}_m) \right\| = \sum_{i=1}^{\mathbf{n}} \sum_{k=1}^{\mathbf{m}} \left[\sum_{j=1}^{\mathbf{n}} [\mathbf{N}]_{ij} \left([\mathbf{\Phi}]_{jk} - [\mathbf{\Phi}_m]_{\mathbf{j}k} \right) \right]^{\mathbf{2}} \tag{2.34}$$

where **N** $= \mathbf{M_a}^{\mathbf{1/2}}$

$\mathbf{M_a}$ is the analytical mass matrix

$\mathbf{\Phi_m}$ is the measured eigenvector

$[\mathbf{N}]_{ij}$, $[\mathbf{\Phi}]_{ij}$, $[\mathbf{\Phi}_m]$ are the (i,j) elements of the matrices **N, Φ,** $\mathbf{\Phi}_m$

m is the number of measured eigenvectors

n is the number of degrees of freedom in the analytical model,

and subjected to the orthogonality condition

$$\mathbf{\Phi}^{\mathbf{T}} \mathbf{M}_a \mathbf{\Phi} = \mathbf{I} \tag{2.35}$$

The Lagrange Multiplier method uses the constraint (2.35) to produce the augmented function to be minimized as (Friswell, 1993)

$$\begin{aligned} J = & \sum_{i,h,j=1}^{n} \sum_{k=1}^{m} \left\{ [\mathbf{N}]_{ij} \left([\mathbf{\Phi}]_{jk} - [\mathbf{\Phi}_m]_{jk} \right) [\mathbf{N}]_{ih} \left([\mathbf{\Phi}]_{hk} - [\mathbf{\Phi}_m]_{hk} \right) \right\} \\ & + \sum_{i,h=1}^{m} \gamma_{jh} \left\{ \sum_{j,k=1}^{n} \left([\mathbf{\Phi}]_{ji} [\mathbf{M}_a] [\mathbf{\Phi}]_{kh} - \delta_{ih} \right) \right\} \end{aligned} \tag{2.36}$$

Where the terms, γ_{jh}, are the Lagrange Multipliers, which are cast into a matrix **Γ**, and the terms δ_{ih} represent errors. The Lagrange Multipliers may be forced to be unique by introducing the constraint of symmetry so that

$$\mathbf{\Gamma} = \mathbf{\Gamma}^{\mathbf{T}} \tag{2.37}$$

Differentiating the augmented function (2.36) with respect to each element of the corrected eigenvector matrix, $[\boldsymbol{\Phi}]_{rs}$, and the following expression is found

$$\boldsymbol{\Phi} = \boldsymbol{\Phi}_m [\mathbf{I} + \boldsymbol{\Gamma}]^{-1} \tag{2.38}$$

which, when substituted back into the orthogonality condition, becomes

$$[\mathbf{I} + \boldsymbol{\Gamma}]^{-1} \boldsymbol{\Phi}_m^{\mathbf{T}} \mathbf{M}_a \boldsymbol{\Phi}_m [\mathbf{I} + \boldsymbol{\Gamma}]^{-1} = \mathbf{I} \tag{2.39}$$

By pre and post multiplying by $(\mathbf{I} + \boldsymbol{\Gamma})$ and taking the square root, it becomes

$$[\mathbf{I} + \boldsymbol{\Gamma}] = [\boldsymbol{\Phi}_m^{\mathbf{T}} \mathbf{M}_a \boldsymbol{\Phi}_m]^{1/2}. \tag{2.40}$$

Finally, substituting equation (2.40) into (2.38), the equation for the corrected eigenvector matrix is

$$\boldsymbol{\Phi} = \boldsymbol{\Phi}_m [\boldsymbol{\Phi}_m^{\mathbf{T}} \mathbf{M}_a \boldsymbol{\Phi}_m]^{-1/2}. \tag{2.41}$$

If it is assumed that the analytical mass matrix is already correct and the eigenvectors are corrected to ensure orthogonality, the stiffness matrix can now be updated. Baruch (1978) found that the updated stiffness matrix, $\mathbf{K}$, can be found to minimize the cost function

$$J = \frac{1}{2} \left\| \mathbf{N}^{-1} (\mathbf{K} - \mathbf{K}_a) \mathbf{N}^{-1} \right\| \tag{2.42}$$

$$J = \frac{1}{2} \sum_{i,j=1}^{n} \left[\sum_{h,k=1}^{n} [\mathbf{N}^{-1}]_{ih} \left([\mathbf{K}]_{hk} - [\mathbf{K}_a]_{hk} \right) [\mathbf{N}^{-1}]_{kj} \right]^{\mathbf{2}} \tag{2.43}$$

where $\mathbf{N} = \mathbf{M}_a^{1/2}$

$[\mathbf{N}^{-1}]_{ij}$, $[\mathbf{K}]_{ij}$, $[\mathbf{K}_a]_{ij}$ are the (i,j) elements of the matrices $\mathbf{N}^{-1}$, $\mathbf{K}$, $\mathbf{K}_a$

and is subject to the two constraints

$$\mathbf{K\Phi} = \mathbf{M}_a\mathbf{\Phi\Lambda} \quad \text{and} \quad \mathbf{K}^{\mathbf{T}} = \mathbf{K}. \tag{2.44}$$

The cost function is then differentiated with respect to the updated stiffness matrix, $[\mathbf{K}]_{rs}$, and results in the following equation

$$\mathbf{M}_a^{-1}(\mathbf{K} - \mathbf{K}_a)\mathbf{M}_a^{-1} + 2\mathbf{\Gamma}_{\mathbf{\Lambda}}\mathbf{\Phi}^{\mathbf{T}} + 2\mathbf{\Gamma}_{\mathbf{\Lambda}} = \mathbf{0} \tag{2.45}$$

where $\mathbf{\Gamma}_{\mathbf{\Lambda}}$ and $\mathbf{\Gamma}_{\mathbf{K}}$ are Lagrange Multipliers.

By calculating the values of the Lagrange Multipliers, substituting them into equation (2.42), and then rearranging equation, the updated stiffness matrix can be found using the following equation

$$\mathbf{K} = \mathbf{K}_a - \mathbf{K}_a\mathbf{\Phi\Phi}^{\mathbf{T}}\mathbf{M}_a - \mathbf{M}_a\mathbf{\Phi\Phi}^{\mathbf{T}}\mathbf{K}_a + \mathbf{M}_a\mathbf{\Phi\Phi}^{\mathbf{T}}\mathbf{K}_a\mathbf{\Phi\Phi}^{\mathbf{T}}\mathbf{M}_a + \mathbf{M}_a\mathbf{\Phi\Lambda\Phi}^{\mathbf{T}}\mathbf{M}_a \tag{2.46}$$

The MATLAB script for this method can be found in Appendix B1.

Using the stiffness updating equation, (2.46), the updated stiffness matrix is found to be

$$\mathbf{K_u} = \begin{bmatrix} 10000 & -5200 & 0 & 0 \\ -5200 & 10000 & -5200 & 0 \\ 0 & -5200 & 10000 & -5200 \\ 0 & 0 & -5200 & 5200 \end{bmatrix} \tag{2.47}$$

Using the updated stiffness matrix, the updated eigenvalues and eigenvectors are

$$\mathbf{L_u} = \begin{bmatrix} 6.56 & 0 & 0 & 0 \\ 0 & 82.28 & 0 & 0 \\ 0 & 0 & 198.03 & 0 \\ 0 & 0 & 0 & 299.81 \end{bmatrix} \tag{2.48}$$

$$\mathbf{V_u} = \begin{bmatrix} -0.38 & 0.98 & 1.00 & -0.65 \\ -0.71 & 0.95 & -0.36 & 1.00 \\ -0.92 & -0.05 & -0.87 & -0.89 \\ -1.00 & -1.00 & 0.68 & 0.36 \end{bmatrix} \tag{2.49}$$

and the eigenvectors have MAC values of

$$\mathbf{MAC} = \begin{Bmatrix} 1.00 \\ 1.00 \\ 1.00 \\ 1.00 \end{Bmatrix}. \tag{2.50}$$

This method reproduces the stiffness and mass matrices that are used to calculate the measured eigenvalues and eigenvectors. For the second case, where the equation (2.46) is used to update the stiffness matrix, reduces to

$$\mathbf{K_u} = \begin{bmatrix} 10581.27 & -5634.48 & -87.47 & 79.10 \\ -5634.48 & 10512.40 & -5512.82 & 62.37 \\ -87.47 & -5512.82 & 10706.91 & -5435.90 \\ 79.10 & 62.37 & -5435.90 & 5251.38 \end{bmatrix} \tag{2.51}$$

it is apparent that the updated stiffness matrix is now filled and no longer physically represents the four-storey model. However, the updated eigenvalue and eigenvectors, equations (2.43) and (2.44), are exactly equivalent to their measured counterparts. The updated eigenvectors appear in normalized form because of the eigenvector correction done in order to ensure mass orthogonalization.

$$\mathbf{L_u} = \begin{bmatrix} 6.56 & 0 & 0 & 0 \\ 0 & 82.28 & 0 & 0 \\ 0 & 0 & 211.08 & 0 \\ 0 & 0 & 0 & 317.62 \end{bmatrix} \tag{2.52}$$

$$\mathbf{V_u} = \begin{bmatrix} -0.39 & 0.95 & -1.00 & 0.65 \\ -0.71 & 0.94 & 0.35 & -1.00 \\ -0.91 & -0.03 & 0.88 & 0.88 \\ -1.00 & -1.00 & -0.65 & -0.35 \end{bmatrix} \tag{2.53}$$

the eigenvectors have MAC values of

$$\mathbf{MAC} = \begin{Bmatrix} 1.00 \\ 1.00 \\ 1.00 \\ 1.00 \end{Bmatrix}. \tag{2.54}$$

Also, updated measured eigenvector

$$\mathbf{V_{mu}} = \begin{bmatrix} -0.032 & 0.073 & 0.085 & -0.055 \\ -0.058 & 0.073 & -0.029 & 0.085 \\ -0.075 & -0.003 & -0.075 & -0.075 \\ -0.082 & -0.078 & 0.055 & 0.029 \end{bmatrix} \tag{2.55}$$

Berman and Nagy (1983) used a similar approach to the one presented by Baruch, however, they used it to update both the mass and stiffness matrices by assuming that the measured eigenvector matrix is correct. The advantage of this method is that it is not necessary to calculate the corrected eigenvectors because the mass matrix is updated in such a manner to ensure the orthogonality of the eigenvectors to the mass matrix.

Given the analytical mass matrix, $\mathbf{M_a}$, and the measured eigenvector matrix, $\mathbf{\Phi}_m$, the following cost function is created in which the updated mass matrix, **M,** is found to minimize the function

$$J = \frac{1}{2}\left\| \mathbf{M_a}^{-1/2}\left(\mathbf{M} - \mathbf{M_a}\right)\mathbf{M_a}^{-1/2} \right\|. \tag{2.56}$$

This function is also subject to the orthogonality constraint

$$\mathbf{\Phi}_m^{\mathbf{T}}\mathbf{M}\mathbf{\Phi}_m = \mathbf{I} \tag{2.57}$$

The cost function J is minimized using the same steps as the cost function containing the corrected stiffness matrix. The result is

$$\mathbf{M_a^{-1}}\left(\mathbf{M}-\mathbf{M_a}\right)\mathbf{M_a^{-1}} + \mathbf{\Phi}_m\mathbf{\Gamma}\mathbf{\Phi}_m^{\mathbf{T}} = \mathbf{0}. \tag{2.58}$$

Combining this equation with that of the orthogonality constraint, (2.50), and the Lagrange Multiplier, the updated mass matrix can be found by adding an updating term, the second term in equation (2.58), to the analytical mass matrix as follows

$$\mathbf{M} = \mathbf{M_a} + \mathbf{M_a}\mathbf{\Phi_m}\overline{\mathbf{M}}_\mathbf{a}^{-1}\left(\mathbf{I}-\overline{\mathbf{M}}_\mathbf{a}\right)\overline{\mathbf{M}}_\mathbf{a}^{-1}\mathbf{\Phi}_m^{\mathbf{T}}\mathbf{M_a} \tag{2.59}$$

where $\overline{\mathbf{M}}_\mathbf{a}^{-1} = \mathbf{\Phi}_m^{\mathbf{T}}\mathbf{M_a}\mathbf{\Phi}_m$.

The updated mass matrix can now be used to calculate the updated stiffness matrix. Since the eigenvector matrix is orthogonal to the newly updated mass matrix, the calculation for the updated stiffness matrix from the previous section can be used; however, the newly acquired updated mass matrix, $\mathbf{M}$, and the measured eigenvector matrix, $\mathbf{\Phi}_m$, will appear in place of the analytical mass matrix, $\mathbf{M_a}$, and the corrected eigenvector matrix, $\mathbf{\Phi}$. So the equation for the updated stiffness matrix becomes

$$\mathbf{K} = \mathbf{K_a} - \mathbf{K_a}\mathbf{\Phi_m}\mathbf{\Phi_m^T}\mathbf{M} - \mathbf{M}\mathbf{\Phi_m}\mathbf{\Phi_m^T}\mathbf{K_a} + \mathbf{M}\mathbf{\Phi_m}\mathbf{\Phi_m^T}\mathbf{K_a}\mathbf{\Phi_m}\mathbf{\Phi_m^T}\mathbf{M} + \mathbf{M}\mathbf{\Phi_m}\mathbf{\Lambda}\mathbf{\Phi_m^T}\mathbf{M}. \tag{2.60}$$

The MATLAB script for this method can be found in Appendix B2.

Using equations (2.59) and (2.60), the updated mass and stiffness matrices are

$$\mathbf{M_u} = \begin{bmatrix} 60 & 0 & 0 & 0 \\ 0 & 60 & 0 & 0 \\ 0 & 0 & 60 & 0 \\ 0 & 0 & 0 & 60 \end{bmatrix} \tag{2.61}$$

$$\mathbf{K_u} = \begin{bmatrix} 10000 & -5200 & 0 & 0 \\ -5200 & 10000 & -5200 & 0 \\ 0 & -5200 & 10000 & -5200 \\ 0 & 0 & -5200 & 5200 \end{bmatrix} \tag{2.62}$$

These are the exact same mass and stiffness matrices as the previous method (see equation (2.45)), and they produce the same eigenvalues and eigenvectors, as seen in (2.63) and (2.64) compared with equations (2.48) and (2.49). The updated eigenvectors, for this case, were not normalized.

$$\mathbf{L_u} = \begin{bmatrix} 6.56 & 0 & 0 & 0 \\ 0 & 82.28 & 0 & 0 \\ 0 & 0 & 198.03 & 0 \\ 0 & 0 & 0 & 299.81 \end{bmatrix} \tag{2.63}$$

$$\mathbf{V_u} = \begin{bmatrix} -0.031 & 0.075 & 0.085 & -0.055 \\ -0.058 & 0.073 & -0.031 & 0.084 \\ -0.075 & -0.004 & -0.073 & -0.075 \\ -0.082 & -0.076 & 0.057 & 0.030 \end{bmatrix} \tag{2.64}$$

the eigenvectors have MAC values of

$$\mathbf{MAC} = \begin{Bmatrix} 1.00 \\ 1.00 \\ 1.00 \\ 1.00 \end{Bmatrix} \tag{2.65}$$

For the second case, the mass and stiffness matrices are updated using equations (2.59) and (2.60) to become

$$\mathbf{M_u} = \begin{bmatrix} 59.61 & 0 & 0.43 & 1.00 \\ 0 & 59.82 & 0.24 & 0.57 \\ 0.43 & 0.24 & 59.53 & 0 \\ 1.00 & 0.57 & 0 & 57.14 \end{bmatrix} \tag{2.66}$$

$$\mathbf{K_u} = \begin{bmatrix} 10534.94 & -5669.21 & -57.17 & 163.14 \\ -5669.21 & 10481.78 & -5486.19 & 135.72 \\ -57.17 & -5486.19 & 10712.37 & -5475.58 \\ 163.14 & 135.72 & -5475.58 & 5114.96 \end{bmatrix} \tag{2.67}$$

Again, the updated matrices become completely filled for the second case. However, since both the mass and stiffness matrices are allowed to be perturbed, they are closer to physically representing the system. The eigenvalues and eigenvectors are found to be

$$\mathbf{L_u} = \begin{bmatrix} 6.45 & 0 & 0 & 0 \\ 0 & 82.00 & 0 & 0 \\ 0 & 0 & 213.52 & 0 \\ 0 & 0 & 0 & 320.14 \end{bmatrix} \tag{2.68}$$

$$\mathbf{V_u} = \begin{bmatrix} -0.37 & 0.94 & -1.00 & -0.65 \\ -0.68 & 0.92 & 0.37 & 1.00 \\ -0.89 & -0.05 & 0.87 & -0.90 \\ -1.00 & -1.00 & -0.66 & 0.38 \end{bmatrix} \tag{2.69}$$

Similar to the results of Baruch’s direct method, the eigenvalues corresponding to those that are not measured are significantly different than what should be expected. However, the MAC values suggest that the eigenvectors are still exact.

$$\mathbf{MAC} = \begin{Bmatrix} 1.00 \\ 1.00 \\ 1.00 \\ 1.00 \end{Bmatrix} \tag{2.70}$$

The Lagrange multiplier methods reproduce the measured eigen-system, however, the results are not physically meaningful, or in other words cause the updated system to lose its physical representation. This is a potential problem for situations where the stiffness and/or mass of a specific degree of freedom is needed, such as in damage detection. These methods are advantageous for systems that contain measured eigenvalue and eigenvectors for every degree of freedom, especially if the physical representation of the mass and stiffness matrices is not of importance. If, however, the measured eigen-system is not complete, the eigenvalues corresponding to the unmeasured degrees of freedom will be updated without control.

2.4 Penalty Function Model Updating Method

The main idea behind the penalty function methods is to optimize a non-linear penalty function to maximize the correlation between the numerical and experimental data. These methods generally employ a truncated Taylor series of modal data as a function of the unknown parameters.

In contrast to the Lagrange multiplier methods, the penalty function method retains the physical representation of the stiffness and mass matrices.

2.4.1 Penalty Function Method

The functions themselves are often non-linear and will only work correctly if an iterative procedure is used. The truncated Taylor series function is as follows:

$$\delta\mathbf{z} = \mathbf{S}_j\ \delta\boldsymbol{\theta} \tag{2.71}$$

where $\delta\boldsymbol{\theta} = \theta - \theta_{\mathbf{j}}$ is the change in the parameters

$\delta\mathbf{z} = \mathbf{z_m} - \mathbf{z_j}$ is the difference between the measured and analytical eigenvalues and eigenvectors

$\mathbf{S_j}$ is the sensitivity matrix.

$$\mathbf{z}_\mathbf{m}^\mathbf{T} = \left(\lambda_{m1}, \boldsymbol{\varphi}_\mathbf{m1}^\mathbf{T}, \lambda_{m2}, \ldots, \lambda_{mr}, \boldsymbol{\varphi}_\mathbf{mr}^\mathbf{T}\right)^\mathbf{T} \tag{2.72}$$

$$\mathbf{z}^\mathbf{T} = \left(\lambda_1, \boldsymbol{\varphi}_\mathbf{1}^\mathbf{T}, \lambda_2, \ldots, \lambda_r, \boldsymbol{\varphi}_\mathbf{r}^\mathbf{T}\right)^\mathbf{T} \tag{2.73}$$

In this equation (2.71) j represents the iteration number; parameter θ_j represents the estimated parameters at iteration j. The parameters being adjusted are up to the user. For example, the non-zero terms in the stiffness and mass matrices could be chosen. The vectors $\mathbf{z}_m$ and $\mathbf{z}_j$ contain the measured eigenvalues and eigenvectors; however, not every eigenvalue or eigenvector is needed for this method to work properly (Miguel, 2006).

2.4.2 Sensitivity Matrix Calculation

The sensitivity matrix $\mathbf{S}_\mathrm{j}$ contains the first derivatives of the eigenvalues and mode shapes with respect to each parameter. For an un-damped system, the first derivative of the eigenvalues was derived by Wittrick (1962) and the first derivative for the eigenvectors was derived by Fox and Kapoor (1968). It is important that the eigenvectors used are mass normalized. In this case, the subscript j represents the j^{th} eigenvalue and eigenvector.

Beginning with the structural eigenproblem

$$\mathbf{K}\boldsymbol{\varphi}_i = \lambda_i \mathbf{M}\boldsymbol{\varphi}_i \tag{2.74}$$

By differentiating the structural eigenproblem with respect to the parameter, θ_j, that are being updated, the following is produced

$$[\mathbf{K} - \lambda_i \mathbf{M}] \frac{\partial \boldsymbol{\varphi}_i}{\partial \theta_j} = -\left[\frac{\partial \mathbf{K}}{\partial \theta_j} - \lambda_i \frac{\partial \mathbf{M}}{\partial \theta_j} - \frac{\partial \lambda_i}{\partial \theta_j} \mathbf{M} \right] \boldsymbol{\varphi}_i . \tag{2.75}$$

The first derivative for of the eigenvalues can be found by pre-multiplying equation (2.75) by the mass normalized $\boldsymbol{\varphi}_i^{\mathrm{T}}$, and using the orthogonality condition as is shown in (2.35)

$$\frac{\partial \lambda_i}{\partial \theta_j} = \boldsymbol{\varphi}_i^{\mathbf{T}} \left[\frac{\partial \mathbf{K}}{\partial \theta_j} - \lambda_i \frac{\partial \mathbf{M}}{\partial \theta_j} \right] \boldsymbol{\varphi}_i . \tag{2.76}$$

Nelson (1976) described a technique that calculates the sensitivities of the eigenvectors using only the j^{th} eigenvalue and eigenvector. Combining equations (2.75) and (2.76) results in the following equation:

$$\left[\frac{\partial \mathbf{K}}{\partial \theta_j} - \lambda_i \frac{\partial \mathbf{M}}{\partial \theta_j} \right] \frac{\partial \boldsymbol{\varphi}_i}{\partial \theta_j} = \mathbf{f}_i \tag{2.77}$$

where $\mathbf{f}_i$ is defined as

$$\mathbf{f}_i = -\left[\frac{\partial \mathbf{K}}{\partial \theta_j} - \lambda_i \frac{\partial \mathbf{M}}{\partial \theta} - \boldsymbol{\varphi}_i^{\mathbf{T}} \left[\frac{\partial \mathbf{K}}{\partial \theta_j} - \lambda_i \frac{\partial \mathbf{M}}{\partial \theta_j} \right] \boldsymbol{\varphi}_i \mathbf{M} \right] \boldsymbol{\varphi}_i . \tag{2.78}$$

The entire eigenvector derivative is broken down into two parts

$$\frac{\partial \boldsymbol{\varphi}_j}{\partial \theta_i} = \mathbf{v}_j + c_j \boldsymbol{\varphi}_i \tag{2.79}$$

where $\mathbf{v}_j$ is equivalent to $\mathbf{f}_j$ and the vector c_j $\boldsymbol{\varphi}_j$ is the homogeneous solution. Then the mass normalization equation (2.35) is differentiated with respect to the parameters, θ_j,

$$\boldsymbol{\varphi}_i^{\mathbf{T}}\mathbf{M}\boldsymbol{\varphi}_i = \mathbf{I} \tag{2.80}$$

When (2.35) is combined with the equation (2.79), the term $\frac{\partial \phi_i}{\partial \theta_j}$ can be eliminated and the result is

$$c_j = -\boldsymbol{\varphi}_i^{\mathbf{T}}\mathbf{M}\mathbf{v}_{\mathbf{j}} - \frac{1}{2}\boldsymbol{\varphi}_i^{\mathbf{T}}\frac{\delta M}{\delta \theta_j}\boldsymbol{\varphi}_i. \tag{2.81}$$

However, since the rank of [**K** – λ_j**M**] = *n-1*, the calculation of $\mathbf{v}_j$ cannot be completed. So, in order to find the vector $\mathbf{v}_j$, one of the terms is set to zero. The calculation becomes

$$\begin{bmatrix} [\mathbf{K}-\lambda_j\mathbf{M}]_{11} & 0 & [\mathbf{K}-\lambda_j\mathbf{M}]_{13} \\ 0 & 1 & 0 \\ [\mathbf{K}-\lambda_j\mathbf{M}]_{31} & 0 & [\mathbf{K}-\lambda_j\mathbf{M}]_{33} \end{bmatrix} \begin{Bmatrix} \mathbf{v_1} \\ v_k \\ \mathbf{v_3} \end{Bmatrix} = \begin{Bmatrix} \mathbf{f_1} \\ 0 \\ \mathbf{f_3} \end{Bmatrix} \tag{2.82}$$

The location of k is chosen where the value of $\left|\{\boldsymbol{\varphi}_i\}_k\right|$ is a maximum. The setting of $v_k = 0$ is offset by the computation of c_j (Friswell, 1993).

Once the first derivatives of the eigenvalues and eigenvectors are calculated, they can be placed in sensitivity matrix. It is important that they are placed in an order that relates the parameter in question to the eigenvalue or eigenvector that it is affecting.

2.4.3 Penalty Function

The linear approximation shown in equation (2.71) can be used to create a cost function by quantifying the error of the predicted measurements as:

$$\boldsymbol{\varepsilon} = \delta\mathbf{z} - \mathbf{S}_{\mathbf{j}}\delta\boldsymbol{\theta} \tag{2.83}$$

Since the error contains the parameters, the Least Squares solution can be found by minimizing the cost function:

$$J = \boldsymbol{\varepsilon}^{\mathbf{T}}\boldsymbol{\varepsilon} = \left\{\delta\mathbf{z} - \mathbf{S}_{\mathbf{j}}\delta\boldsymbol{\theta}\right\}^{T}\left\{\delta\mathbf{z} - \mathbf{S}_{\mathbf{j}}\delta\boldsymbol{\theta}\right\} \tag{2.84}$$

which can be simply solved as:

$$\boldsymbol{\theta}_{\mathbf{j+1}} = \boldsymbol{\theta}_{\mathbf{j}} + \left[\mathbf{S}_{\mathbf{j}}^{\mathbf{T}}\mathbf{S}_{\mathbf{j}}\right]^{-1}\mathbf{S}_{\mathbf{j}}^{\mathbf{T}}\left(\mathbf{z}_{\mathbf{m}} - \mathbf{z}_{\mathbf{j}}\right). \tag{2.85}$$

However, it should be noted that this equation is only valid when there are more eigenvalue and eigenvector measurements than the number of parameters. If there are more parameters than measurements, the following equation must be used

$$\boldsymbol{\theta}_{\mathbf{j+1}} = \boldsymbol{\theta}_{\mathbf{j}} + \mathbf{S}_{\mathbf{j}}^{\mathbf{T}}\left[\mathbf{S}_{\mathbf{j}}\mathbf{S}_{\mathbf{j}}^{\mathbf{T}}\right]^{-1}\left(\mathbf{z}_{\mathbf{m}} - \mathbf{z}_{\mathbf{j}}\right). \tag{2.86}$$

The MATLAB script for this method can be found in Appendix B4.

It should be noted that this example was performed with only the measured eigenvalues, and does not include the use of the measured eigenvectors; this is because the calculations for the first derivative of the eigenvectors is computationally difficult. Since this was a smaller example, it was possible to utilize MATLAB to complete several iterations. For the results below, 1000 iterations were used, however, the solution is found to converge after only 5 iterations.

The updated stiffness matrix converges to

$$\mathbf{K}_{\mathbf{u}} = \begin{bmatrix} 10119.28 & -5137.96 & 0 & 0 \\ -5137.96 & 10117.23 & -5142.82 & 0 \\ 0 & -5142.82 & 10115.41 & -5147.92 \\ 0 & 0 & -5147.92 & 4848.09 \end{bmatrix}. \tag{2.87}$$

Using the newly updated stiffness matrix, the eigenvalues and eigenvectors are found to be

$$\mathbf{L_u} = \begin{bmatrix} 6.56 & 0 & 0 & 0 \\ 0 & 82.28 & 0 & 0 \\ 0 & 0 & 198.03 & 0 \\ 0 & 0 & 0 & 299.81 \end{bmatrix} \tag{2.88}$$

$$\mathbf{V_u} = \begin{bmatrix} -0.029 & 0.075 & 0.085 & -0.055 \\ -0.055 & 0.075 & -0.029 & 0.085 \\ -0.074 & 0.0013 & -0.075 & -0.075 \\ -0.086 & -0.074 & 0.055 & 0.029 \end{bmatrix} \tag{2.89}$$

and the eigenvectors have MAC values of

$$\mathbf{MAC} = \begin{Bmatrix} 1.00 \\ 1.00 \\ 1.00 \\ 1.00 \end{Bmatrix}. \tag{2.90}$$

Despite using only the measured eigenvalues in the iteration process, there was a larger error in the updated eigenvalues than in the eigenvectors, which were nearly exactly re-produced.

The penalty function method was used to update the stiffness matrix for the second case, in which only two of the measured eigenvalues were used. This method was able to reproduce the two measured eigenvalues exactly, however, the third and fourth eigenvalues were incorrect. The main advantage the penalty method has over the Lagrange multiplier methods is that the stiffness method retains its form and, hence, its physical representation, as seen in equation (2.91) .

$$\mathbf{K_u} = \begin{bmatrix} 10597.00 & -5786.85 & 0 & 0 \\ -5786.85 & 10591.80 & -5415.15 & 0 \\ 0 & -5415.15 & 10777.39 & -5426.53 \\ 0 & 0 & -5426.53 & 5185.85 \end{bmatrix} \tag{2.91}$$

$$\mathbf{L_u} = \begin{bmatrix} 6.56 & 0 & 0 & 0 \\ 0 & 82.28 & 0 & 0 \\ 0 & 0 & 210.92 & 0 \\ 0 & 0 & 0 & 319.44 \end{bmatrix} \tag{2.92}$$

$$\mathbf{V_u} = \begin{bmatrix} -0.033 & 0.074 & 0.082 & -0.057 \\ -0.058 & 0.073 & -0.029 & 0.085 \\ -0.073 & -0.004 & -0.077 & -0.073 \\ -0.083 & -0.076 & 0.056 & 0.028 \end{bmatrix} \tag{2.93}$$

Similar to the Lagrange multiplier methods, even though the updated third and fourth eigenvalues did not match the measured values, the updated and measured eigenvectors were exactly reproduced, as confirmed by the following MAC values:

$$\mathbf{MAC} = \begin{Bmatrix} 1.00 \\ 1.00 \\ 1.00 \\ 1.00 \end{Bmatrix}. \tag{2.94}$$

The penalty function method contains much more user interaction, as the user must specify the terms in the stiffness matrix to be updated and their sensitivity, which can become very difficult for larger systems, especially if the user knowledge of the FE model is limited.

However, the main advantage is the fact that the form of the stiffness matrix is retained.

3.0 Control-based Model Updating Methods

This chapter describes four control-based model updating methods: (i) the state-space eigen-structure assignment method, (ii) the quadratic pencil method, (iii) the constrained eigen-structure assignment method, (iv) and the altered constrained eigen-structure method. A numerical method is also presented, to illustrate the key concepts and steps involved in each of these methods. All of the methods presented in this chapter are derived using control theory, which provides the ability to update a subset of modal parameters while leaving the rest unchanged. This ability is advantageous since it is very difficult to extract the complete eigen-structure in large scale structures.

3.1 Control Theory

The basic idea underlying control-based model updating methods is the use of the concepts of feed-back control to simultaneously assign the system eigenvalues and eigenvectors to desired locations. Feed-back in terms of control theory is when a controller (gain matrix) is used to adjust the properties of a defined system.

Unlike actual feed-back systems that utilize active forces to achieve this placement, in this study, this feed-back is fictitious and is only a mathematical procedure. Consider a dynamical system in state-space as:

$$\dot{\mathbf{x}}(t) = \mathbf{A}\mathbf{x}(t) + \mathbf{B}\mathbf{u}(t) \tag{3.1}$$

$$\mathbf{y}(t) = \mathbf{C}\mathbf{x}(t) \tag{3.2}$$

where **A** is the state matrix

B is the input matrix

x(t) is the state vector

C is the output matrix

u(*t*) is an external force vector, or the feed-back force.

y(t) is the response vector.

In equation (3.1), **A** is the state matrix, **B** determines the locations of the feedback forces represented by **u**(*t*). The main idea of the control-based methods is to determine the control vector **u**(*t*), which is fictitious, that will place the eigenvalues and eigenvectors of **A** in their desired locations, a process known as eigen-structure assignment.

For building structures, the state space matrix is constructed using the stiffness, **K**, mass, **M**, and damping, **D**, matrices as follows

$$\mathbf{A} = \begin{bmatrix} \mathbf{0} & \mathbf{I} \\ -\mathbf{M}^{-1}\mathbf{K} & -\mathbf{M}^{-1}\mathbf{D} \end{bmatrix} \tag{3.3}$$

The eigenvalue problem for state-space is set up similar to that of the quadratic case (Andry, 1983). The eigenvalues are found by solving the characteristic equation

$$\det[\lambda\mathbf{I} - \mathbf{A}] = 0 \tag{3.4}$$

In order to accomplish the eigen-structure assignment the form of the control vector **u**(*t*) is first chosen; (a) full-state feed-back with **u**(*t*) of the form, **u**(*t*) = -**Kx**(*t*), where **K** is a gain matrix and **x**(*t*) is a state vector. Since all the states are used in **x**(*t*), it is known as full-state feed-back, (b) output feed-back, with **u**(*t*) = -**Ky**(*t*).

In the following examples, only case (a) will be considered. For this case, the state equations in (3.1) become:

$$\dot{\mathbf{x}}(t) = \mathbf{A}\mathbf{x}(t) + \mathbf{B}(-\mathbf{K}\mathbf{x}(t)) \tag{3.5}$$

and the updated state matrix is defined as

$$\mathbf{A} - \mathbf{BK}. \tag{3.6}$$

The objective is to find a gain matrix, **K**, such that the updated state matrix has the same eigen-structure as the measured values.

3.2 Numerical Example

The numerical example used for this section is a five degree of freedom lumped-mass structure with mass, stiffness, and damping matrices as follows:

$$\mathbf{M} = \begin{bmatrix} 5897 & 0 & 0 & 0 & 0 \\ 0 & 5897 & 0 & 0 & 0 \\ 0 & 0 & 5897 & 0 & 0 \\ 0 & 0 & 0 & 5897 & 0 \\ 0 & 0 & 0 & 0 & 5897 \end{bmatrix} kg \tag{3.7}$$

$$\mathbf{K} = \begin{bmatrix} 62825 & -29093 & 0 & 0 & 0 \\ -29093 & 57714 & -28621 & 0 & 0 \\ 0 & -28621 & 53575 & -24954 & 0 \\ 0 & 0 & -24954 & 44013 & -19059 \\ 0 & 0 & 0 & -19059 & 19059 \end{bmatrix} \times 1000\ N/m \tag{3.8}$$

$$\mathbf{D} = \begin{bmatrix} 125 & -58 & 0 & 0 & 0 \\ -58 & 115 & -57 & 0 & 0 \\ 0 & -57 & 107 & -50 & 0 \\ 0 & 0 & -50 & 88 & -38 \\ 0 & 0 & 0 & -38 & 38 \end{bmatrix} \times 1000\ N\,s/m \tag{3.9}$$

Using equation (3.3), the state matrix can be calculated as

$$
\mathbf{A} = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \\ -10654 & 4934 & 0 & 0 & 0 & -21 & 10 & 0 & 0 & 0 \\ 4934 & -9787 & 4854 & 0 & 0 & 10 & -19 & 10 & 0 & 0 \\ 0 & 4854 & -9085 & 4232 & 0 & 0 & 10 & -18 & 8 & 0 \\ 0 & 0 & 4232 & -7464 & 3232 & 0 & 0 & 8 & -15 & 6 \\ 0 & 0 & 0 & 3232 & -3232 & 0 & 0 & 0 & 6 & -6 \end{bmatrix} \tag{3.10}
$$

To simulate a system identification procedure that would typically occur in the field, the state matrix **A** was arbitrarily perturbed. The perturbed **A** matrix was then assumed to represent "actual" conditions whose eigenvalues and eigenvectors represent the desired values. The original and desired eigenvalues and eigenvectors are shown in Table 1, Table 2, and Table 3 below.

Table 1: Actual and Desired Eigenvalues

Eigenvalues (desired)	Eigenvalues (actual)
-0.78 - 19.88i	-0.40 - 20.10i
-0.78 + 19.88i	-0.40 + 20.10i
-2.95 - 54.19i	-2.99 - 54.68i
-2.95 + 54.19i	-2.99 + 54.68i
-7.15 - 84.30i	-7.29 - 85.25i
-7.15 + 84.30i	-7.29 - 85.25i
-11.99 - 108.68i	-12.21 - 109.92i
-11.99 + 108.68i	-12.21 + 109.92i
-16.86 - 129.07i	-17.21 - 130.27i
-16.86 + 129.07i	-17.21 + 130.27i

In the control-based model updating methods, arbitrarily assigning the eigenvectors is not possible, the desired eigenvectors must be based on the concept of assignable eigenvectors (Andry, 1983). For this thesis, only the eigenvalues will be assigned and the procedure is described in the following section.

Table 2: Eigenvectors

Eigenvectors (desired)									
-0.01 + 0.0028i	-0.008 - 0.0028i	0.62	0.62	0.039 + 0.028i	0.039 - 0.028i	0.0028 - 0.25i	0.0028 + 0.25i	0.025 + 0.076i	0.025 - 0.076i
0.017 - 0.013i	0.017 + 0.013i	-0.28 - 0.10i	-0.28 + 0.10i	-0.018 - 0.12i	-0.0178 + 0.12i	-0.012 - 0.59i	-0.012 + 0.59i	0.027 + 0.072i	0.027 - 0.072i
-0.018 - 0.015i	-0.018 + 0.015i	0.084 - 0.016i	0.084 + 0.016i	-0.0099 - 0.66i	-0.0099 + 0.66i	-0.0043 + 0.085i	-0.0043 - 0.085i	0.034 - 0.040i	0.034 + 0.040i
-0.047 - 0.052i	-0.047 + 0.052i	0.060 - 0.51i	0.060 + 0.51i	-0.13 + 0.067i	-0.13 - 0.067i	-0.065 - 0.025i	-0.065 + 0.025i	0.012 - 0.44i	0.012 + 0.44i
0.021 + 0.11i	0.021 - 0.11i	0.014 + 0.47i	0.014 - 0.47i	-0.073 + 0.032i	-0.073 - 0.032i	0.11 - 0.16i	0.11 + 0.16i	-0.0011 - 0.45i	-0.0011 + 0.45i
0.024 + 0.021i	0.024 - 0.021i	-0.019 - 0.050i	-0.019 + 0.050i	0.71	0.71	-0.0074 - 0.029i	-0.0074 + 0.029i	0.072 - 0.13i	0.072 + 0.13i
0.032 + 0.0095i	0.032 - 0.0095i	0.011 + 0.13i	0.011 - 0.13i	-0.028 + 0.015i	-0.028 - 0.015i	-0.73	-0.73	0.067 - 0.075i	0.067 + 0.075i
0.016 - 0.036i	0.016 + 0.036i	-0.012 + 0.015i	-0.012 - 0.015i	-0.11 + 0.027i	-0.11 - 0.027i	0.099 - 0.012i	0.099 + 0.012i	0.74	0.74
0.018 - 0.68i	0.018 + 0.68i	-0.0046 + 0.12i	-0.0046 - 0.12i	0.0085 + 0.040i	0.0085 - 0.040i	0.006 - 0.018i	0.006 + 0.018i	-0.023 - 0.056i	-0.023 + 0.056i
-0.71	-0.71	-0.011 + 0.076i	-0.011 - 0.076i	0.045 + 0.0013i	0.045 - 0.0013i	-0.033 - 0.015i	-0.032 + 0.015i	0.025 + 0.008i	0.025 - 0.008i

Table 3: Eigenvectors

Eigenvectors (actual)									
0.0005 + 0.0037i	0.0005 - 0.0037i	-0.0006 - 0.005i	-0.0006 + 0.005i	0.0005 + 0.006i	0.0005 - 0.006i	0.0004 + 0.0066i	0.0004 - 0.0066i	-0.0001 - 0.007i	-0.0001 + 0.007i
-0.0007 - 0.005i	-0.0007 + 0.005i	0.0002 + 0.0017i	0.0002 - 0.0017i	0.0003 + 0.0041i	0.0003 - 0.0041i	0.0006 + 0.010i	0.0006 - 0.010i	-0.0003 - 0.015i	-0.0003 + 0.015i
0.0005 + 0.0039i	0.0005 - 0.0039i	0.0005 + 0.0045i	0.0005 - 0.0045i	-0.0003 - 0.004i	-0.0003 + 0.004i	0.0004 + 0.0076i	0.0004 - 0.0076i	-0.0004 - 0.022i	-0.0004 + 0.022i
-0.0002 - 0.002i	-0.0002 + 0.002i	-0.0006 - 0.005i	-0.0006 + 0.005i	-0.0005 - 0.006i	-0.0005 + 0.006i	-0 - 0.0008i	-0 + 0.0008i	-0.0005 - 0.027i	-0.0005 + 0.027i
0.0001 + 0.0004i	0.0001 - 0.0004i	0.0002 + 0.0019i	0.0002 - 0.0019i	0.0004 + 0.005i	0.0004 - 0.005i	-0.0006 - 0.011i	-0.0006 + 0.011i	-0.0006 - 0.031i	-0.0006 + 0.031i
-0.49 + 0.0004i	-0.49 - 0.0004i	0.58	0.58	-0.52 - 0.0004i	-0.52 + 0.0004i	-0.36 - 0.0001i	-0.36 + 0.0001i	0.15	0.15
0.66	0.66	-0.19 - 0.001i	-0.19 + 0.001i	-0.35	-0.35	-0.56 + 0.0001i	-0.56 - 0.0001i	0.30	0.30
-0.51 - 0.0007i	-0.51 + 0.0007i	-0.450 + 0.0006i	-0.50 - 0.0006i	0.35 + 0.0004i	0.35 - 0.0004i	-0.42 + 0.0002i	-0.42 - 0.0002i	0.44	0.44
0.24 + 0.0008i	0.24 - 0.0008i	0.58	0.58	0.55	0.55	0.044	0.044	0.55	0.55
-0.055 - 0.0002i	-0.055 + 0.0002i	-0.21 + 0.0001i	-0.21 - 0.0001i	-0.43 - 0.0001i	-0.43 + 0.0001i	0.61	0.61	0.63	0.63

3.3 State-Space Eigen-structure Assignment Method

The use of eigen-structure assignment has been used for many years in the fields of Mechanical and Electrical Engineering. The use of this method in Civil Engineering is relatively uncommon.

In general, control-based methods can be classified as methods that operate directly on the state matrices and those that operate on the system properties directly. For the state space method, the basic equations can be written as:

$$\dot{\mathbf{x}}(t) = \mathbf{A}\mathbf{x}(t) + \mathbf{B}\mathbf{u}(t) \tag{3.11}$$

$$\mathbf{y}(t) = \mathbf{C}\mathbf{x}(t) \tag{3.12}$$

where **x**, **u**, and **y** are the state, control ,and output matrices; **A, B,** and **C** are real constant matrices; and the rank of **B** and **C** are not 0. It is also important that the system is controllable (Andry, 1983), is defined as:

$$\text{rank}\left[\mathbf{B} \quad \mathbf{AB} \quad \ldots \quad \mathbf{A^{n-1}B}\right] = n \tag{3.13}$$

when the size of **A** is n x n. As described in section 4, the state matrix, **A**, contains the system information, **B**, is the feed-back gain matrix that needs to be determined.

3.3.1 Theory

Given *2n* measured complex eigenvalues and *2n* measured complex eigenvectors, the full-state problem calculates a feedback gain matrix **K** that updates the state matrix:

$$\mathbf{A'}_{n\times n} = \mathbf{A}_{n\times n} - \mathbf{B}_{n\times m}\mathbf{K}_{m\times n} \tag{3.14}$$

such that the closed loop eigenvalues and eigenvectors correspond exactly to the *2n* measured eigenvalues and eigenvectors.

Moore (1976) derived a procedure that produces a matrix **K** that will update the state matrix, **A**, to exactly match the measured eigenvalues and eigenvectors. To begin,

$$\mathbf{S}_\lambda = \left[\boldsymbol{\lambda}_d\mathbf{I} - \mathbf{A} \;\middle|\; \mathbf{B}\right] \quad \text{and} \quad \mathbf{R}_\lambda = \begin{bmatrix} \mathbf{N}_\lambda \\ \mathbf{M}_\lambda \end{bmatrix} \tag{3.15}$$

are defined, where the columns of $\mathbf{R}_\lambda$ from a basis for the nullspace of $\mathbf{S}_\lambda$. It should be noted that $\mathbf{N}_{\lambda^*} = \mathbf{N}_\lambda^*$ where the asterisk represents complex conjugates. The eigenproblem, which was defined as (1.5) for the second order equation, can be defined as

$$(\lambda_i \mathbf{I} - \mathbf{A})\mathbf{v_i} = \mathbf{BKv_i} \tag{3.16}$$

for the state space equation, and rearranged to give:

$$\left[\lambda_i \mathbf{I} - \mathbf{A} \;\middle|\; \mathbf{B}\right]\begin{bmatrix} \mathbf{v_i} \\ -\mathbf{Kv_i} \end{bmatrix} = \mathbf{0} \tag{3.17}$$

It follows from the comparison of equations (3.15) and (3.17) that $\mathbf{v}_i$ spans $\mathbf{N}_\lambda$. If that is the case, then there is a vector $\mathbf{z}_i$ such that

$$\mathbf{v_i} = \mathbf{N}_{\lambda_i}\mathbf{z_i} \tag{3.18}$$

Combining equations (3.15) and (3.17) we find that:

$$(\lambda_i \mathbf{I} - \mathbf{A})\mathbf{N}_{\lambda_i} + \mathbf{BM}_{\lambda_i} = \mathbf{0} \tag{3.19}$$

Then by multiplying the vector $\mathbf{z}_i$

$$(\lambda_i \mathbf{I} - \mathbf{A})\mathbf{N}_{\lambda_i}\mathbf{z_i} + \mathbf{BM}_{\lambda_i}\mathbf{z_i} = \mathbf{0} \tag{3.20}$$

and using the relationship in equation (3.18) the following equation is produced

$$(\lambda_i \mathbf{I} - \mathbf{A})\mathbf{v_i} + \mathbf{BM}_{\lambda_i}\mathbf{z_i} = \mathbf{0} \tag{3.21}$$

Using the parallels between equations (3.19) and (3.21), $\mathbf{K}$ can be found so that

$$-\mathbf{M}_{\lambda_i}\mathbf{z_i} = \mathbf{Kv_i} \tag{3.22}$$

Then

$$\left[\lambda_i \mathbf{I} - (\mathbf{A} + \mathbf{BK})\right]\mathbf{v_i} = 0 \tag{3.23}$$

and by solving

$$\mathbf{K} = -\mathbf{M}_{\lambda_i}\mathbf{z_i}\left[\mathbf{v_i}\right]^{-1} \tag{3.24}$$

the feedback matrix can now be substituted into equation (3.24) to produce a new state matrix with the measured eigenstructure.

Since the eigenvalues and eigenvectors will be complex, before calculating the feedback gain matrix, both sides of equation (3.24) should be multiplied by:

$$\begin{bmatrix} \frac{1}{2} & -\frac{i}{2} & 0 \\ \frac{1}{2} & \frac{i}{2} & 0 \\ 0 & 0 & \ddots \end{bmatrix} \tag{3.25}$$

to eliminate all of the complex terms (Moore, 1976).

The MATLAB script for this method can be found in Appendix B4.

The state space eigen-structure assignment method was used to update the numerical example presented in section 3.2. Since this method works directly in state space form, there was no need to manipulate the example in any way. The updated and desired eigenvalues are as follows:

Table 4: State-space ESA Results

Eigenvalues (desired)	**Eigenvalues (updated)**
-0.78 - 19.88i	-0.78 - 19.88i
-0.78 + 19.88i	-0.78 + 19.88i
-2.95 - 54.19i	-2.95 - 54.19i
-2.95 + 54.19i	-2.95 + 54.19i
-7.15 - 84.30i	-7.15 - 84.30i
-7.15 + 84.30i	-7.15 + 84.30i
-11.99 - 108.68i	-11.99 - 108.68i
-11.99 + 108.68i	-11.99 + 108.68i
-16.86 - 129.07i	-16.86 - 129.07i
-16.86 + 129.07i	-16.86 + 129.07i

The method was able to recreate the desired eigenvalues exactly.

Also, the frequency response of the updated state matrix in comparison to that of the identified state matrix **A** is shown in Figure 6. It can be seen that the updated state matrix is, in fact, a mathematical representation of the measured structure.

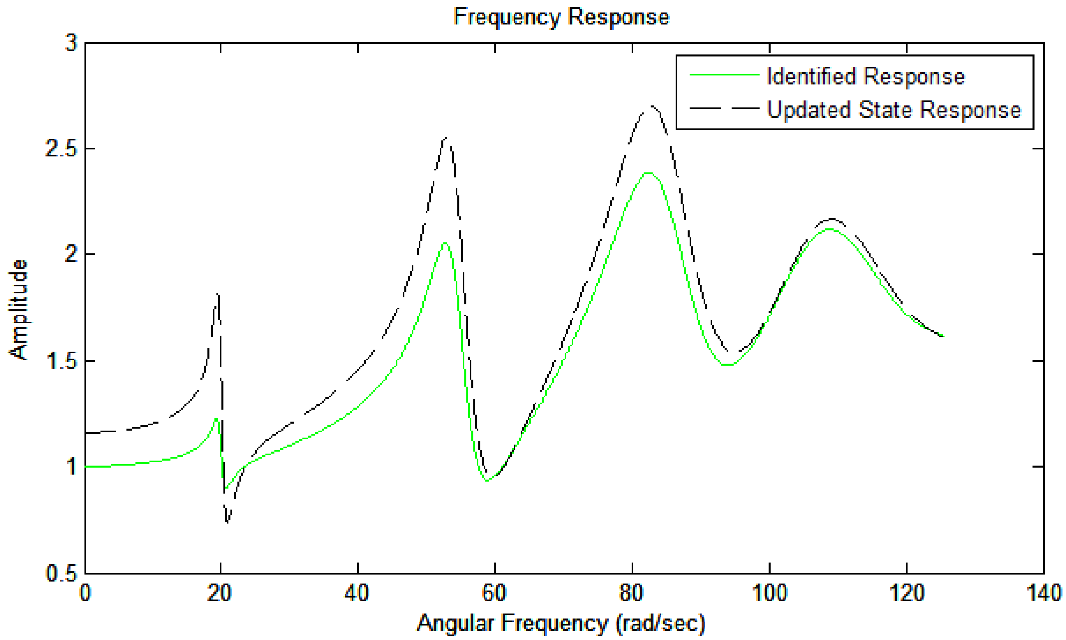

Figure 6: Frequency Responses Using State Space ESA Method

The peaks of the frequency responses correspond to the desired angular frequencies; however, the amplitudes are not the same. This can be attributed to the effect of damping in the updated system.

If only four for the eigenvalues are available; in this case, the first, second, fifth, and sixth eigenvalues; those eigenvalues are updated exactly. However, the remaining updated eigenvalues vary slightly from desired eigenvalues.

Table 5: State-space ESA Results for Partial Identification

Eigenvalues (desired)	Eigenvalues (updated)
-0.78 - 19.88i	-0.78 - 19.88i
-0.78 + 19.88i	-0.78 + 19.88i
-2.95 - 54.19i	-2.88 - 50.30i
-2.95 + 54.19i	-2.88 + 50.30i
-7.15 - 84.30i	-7.15 - 84.30i
-7.15 + 84.30i	-7.15 + 84.30i
-11.99 - 108.68i	-11.68 - 110.25i
-11.99 + 108.68i	-11.68 + 110.25i
-16.86 - 129.07i	-16.75 – 130.48i
-16.86 + 129.07i	-16.75 + 130.48

The frequency response for the partially identified system, seen in Figure 7, shows the first few modes matching almost exactly. However, it can be seen that the peaks of the higher modes occur at different frequencies. So, by using partial identification on the state space method, the frequency response can be re-produced such that they are closer to the identified response than the full identification, however, the location of the peaks is sacrificed.

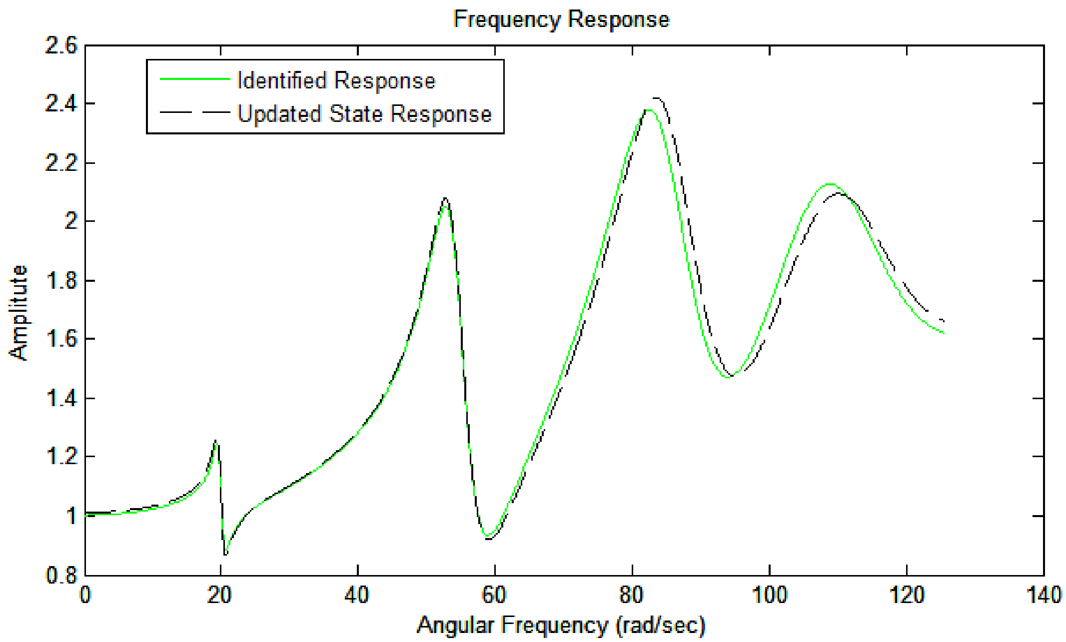

Figure 7: Frequency Responses Using State Space ESA Method (Partial Identification)

3.4 Quadratic Pencil Method

This form of eigenstructure assignment was introduced by Datta (2000) and is similar to the one described in Andry and Shapiro (1983). The procedure is similar; however, the main difference is that the adjustments are performed directly on a second order differential equation (3.26). This leads to the quadratic eigenvalue problem, which is defined by the equation (3.27)

$$\mathbf{M\ddot{x}} + \mathbf{C\dot{x}} + \mathbf{Kx} = \mathbf{0} \tag{3.26}$$

$$\mathbf{MY\Lambda}^2 + \mathbf{CY\Lambda} + \mathbf{KY} = \mathbf{0} \tag{3.27}$$

where **Y** is an *n* x *2n* eigenvector matrix and

Λ is a *2n* x *2n* eigenvalue matrix.

This dynamic equation can be modified by applying a control force **Bu**(t), where **B** is an *n* x *m* matrix and **u**(t) is a *m* x *1* vector which signifies the feed-back force which is defined in equation (3.30), where **F** and **G** are *n* x *m* matrices, called state feedback control matrices. If these definitions are introduced into the eigenvalue problem, it results in equation (3.30). The steps involved are as follows:

$$\mathbf{M\ddot{x}} + \mathbf{C\dot{x}} + \mathbf{Kx} = \mathbf{Bu}(t) \tag{3.28}$$

$$\mathbf{u}(t) = \mathbf{F^T\dot{x}} + \mathbf{G^T x} \tag{3.29}$$

$$\mathbf{MY\Lambda^2} + \left(\mathbf{C} - \mathbf{BF^T}\right)\mathbf{Y\Lambda} + \left(\mathbf{K} - \mathbf{BG^T}\right)\mathbf{Y} = \mathbf{0} \tag{3.30}$$

This will form the basis of the eigenstructure assignment method. Now, given

$$\mathbf{Y} = \begin{pmatrix} \mathbf{Y'_1} & \mathbf{Y_2} \end{pmatrix} \text{ and } \mathbf{\Lambda} = \begin{pmatrix} \mathbf{\Lambda'_1} & \mathbf{0} \\ \mathbf{0} & \mathbf{\Lambda_2} \end{pmatrix} \tag{3.31}$$

where $\mathbf{Y'_1}$ is an n x m matrix of measured eigenvectors,

$\mathbf{Y_2}$ is an n x ($2n$-m) matrix of the remaining unmeasured eigenvectors,

$\mathbf{\Lambda'_1}$ is an m x m matrix of measured eigenvalues, and

$\mathbf{\Lambda_2}$ is an ($2n$-m) x ($2n$-m) of the remaining unmeasured eigenvalues.

The values of $\widehat{\mathbf{B}}^{T}$, $\widehat{\mathbf{F}}^{\mathbf{T}}$, and $\widehat{\mathbf{G}}^{\mathbf{T}}$must be found to satisfy (Datta, 2000)

$$\mathbf{MY\Lambda^2} + \left(\mathbf{C} - \mathbf{\hat{B}\hat{F}^T}\right)\mathbf{Y\Lambda} + \left(\mathbf{K} - \mathbf{\hat{B}\hat{F}^T}\right)\mathbf{Y} = \mathbf{0.} \tag{3.32}$$

Datta (2000) defined:

$$\mathbf{\hat{B}} = \mathbf{MY'_1\,\Lambda'^2_1} + \mathbf{DY'_1\,\Lambda'_1} + \mathbf{KY'_1} \tag{3.33}$$

$$\mathbf{\hat{F}} = \mathbf{MY'_1\,\Lambda_1 Z_1^{-1}} \tag{3.34}$$

$$\mathbf{\hat{G}} = -\mathbf{KY'_1\,Z_1^{-1}} \tag{3.35}$$

Where

$$\mathbf{Z_1} = \mathbf{\Lambda'_1\,Y'^T_1\,MY_1\Lambda_1} - \mathbf{Y'^T_1\,KY_1} \tag{3.36}$$

$\mathbf{\Lambda_1}$	is an m x m matrix of original eigenvalues that are to be reassigned
$\mathbf{Y_1}$	is an n x m matrix of original eigenvectors that are to be reassigned.

and

$$\mathbf{H} = \hat{\mathbf{B}} \left[\hat{\mathbf{F}}^{\mathbf{T}} \middle| \hat{\mathbf{G}}^{\mathbf{T}} \right] \tag{3.37}$$

is defined.

In order to solve **B**, **F**, and **G**, a singular value decomposition (SVD) must be performed on **H**. The compact SVD produces three matrices; **U**, **Σ**, and **V** and appears as such;

$$\mathbf{U}\,\mathbf{\Sigma}\,\mathbf{V}^{\mathbf{T}} = \mathbf{H} \tag{3.38}$$

From which, **B** is taken to be the product of **U** and **Σ**, and the first n rows of **V** are taken to be **F,** and the last n rows are taken to be **G**.

The MATLAB script for this method can be found in Appendix B5.

The quadratic pencil method is used to update the numerical example presented in section 3.2. However, unlike the state space ESA method, the quadratic pencil method can only update the mass and damping matrices. This requires the user to ensure that the initial system is given in terms of the mass, stiffness, and damping matrices. The updated system, however, can be re-organized into a state matrix; the updated state matrix is as follows:

$$\mathbf{A} = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \\ -10320 & 4784 & -0.8 & -1 & -0.3 & -22 & 10 & 0.1 & 0.02 & 0.01 \\ 4857 & -9631 & 4779 & -0.9 & -0.9 & 10 & -20 & 10 & 0.09 & 0.06 \\ 0.1 & 4819 & -9013 & 4197 & 0.3 & 0.09 & 10 & -18 & 9 & 0.2 \\ 2 & -3 & 4134 & -7287 & 3156 & 0.1 & 0.06 & 9 & -15 & 7 \\ 2 & -2 & -0.5 & 3181 & -3178 & 0.2 & 0.08 & 0.1 & 7 & -6 \end{bmatrix} \tag{3.39}$$

The updated state matrix using the quadratic pencil method appears more similar to the analytical state matrix than the updated state matrix provided by the state space ESA method; this is because the mass and damping matrices were updated individually. Figure 8 displays the frequency responses of both the identified state matrix and the updated state matrix. Table 6 displays the desired and updated eigenvalues.

Table 6: Quadratic Pencil Method Results

Eigenvalues (desired)	Eigenvalues (updated)
-0.78 - 19.88i	-0.68 - 19.88i
-0.78 + 19.88i	-0.68 + 19.88i
-2.95 - 54.19i	-2.83 - 54.19i
-2.95 + 54.19i	-2.83 + 54.19i
-7.15 - 84.30i	-7.41 - 84.28i
-7.15 + 84.30i	-7.41 + 84.28i
-11.99 - 108.68i	-12.40 - 108.67i
-11.99 + 108.68i	-12.40 + 108.67i
-16.86 - 129.07i	-17.44 - 129.04i
-16.86 + 129.07i	-17.44 + 129.04i

All of the updated eigenvalues vary slightly from their desired counterparts. This can be explained by the fact that the quadratic pencil method require the use of desired eigenvectors; however, the measured eigenvectors are not in a form that can be used. Instead, a small variation of the analytical eigenvectors is used.

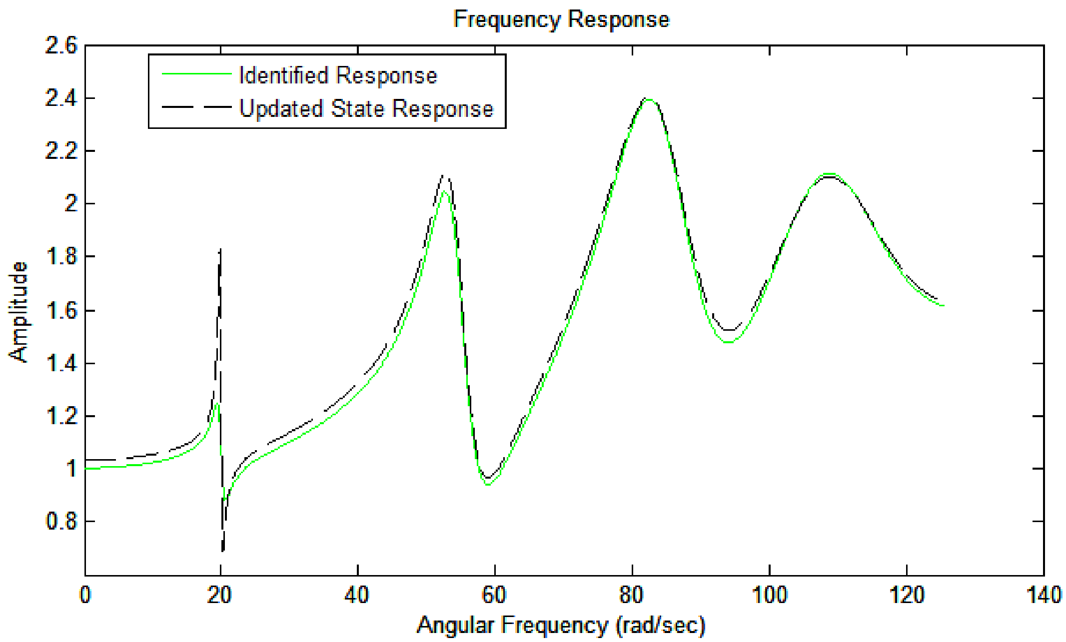

Figure 8: Frequency Responses Using Quadratic Pencil Method

Although the magnitudes of the updated state response differ from the identified state matrix, they are much closer than the magnitudes seen in the state space ESA method.

When using the quadratic pencil method in the case of partial identification, Table 7 shows that the updated eigenvalues, again, vary slightly from the desired eigenvalues.

Table 7: Quadratic Pencil Method Results for Partial Identification

Eigenvalues (desired)	Eigenvalues (updated)
-0.78 - 19.88i	-0.73 - 19.89i
-0.78 + 19.88i	-0.73 + 19.89i
-2.95 - 54.19i	-3.02 - 54.19i
-2.95 + 54.19i	-3.02 + 54.19i
-7.15 - 84.30i	-7.29 - 85.23i
-7.15 + 84.30i	-7.29 + 85.23i
-11.99 - 108.68i	-12.21 - 109.92i
-11.99 + 108.68i	-12.21 + 109.92i
-16.86 - 129.07i	-17.21 - 130.27i
-16.86 + 129.07i	-17.44 + 130.27i

These results are similar to the small errors found for the complete identification case. Figure 9 shows the frequency response for the updated system using partial identification. Similar to the partial identification case for the state space ESA, the updated state response is closer in amplitude but contains more error in the location of the peaks.

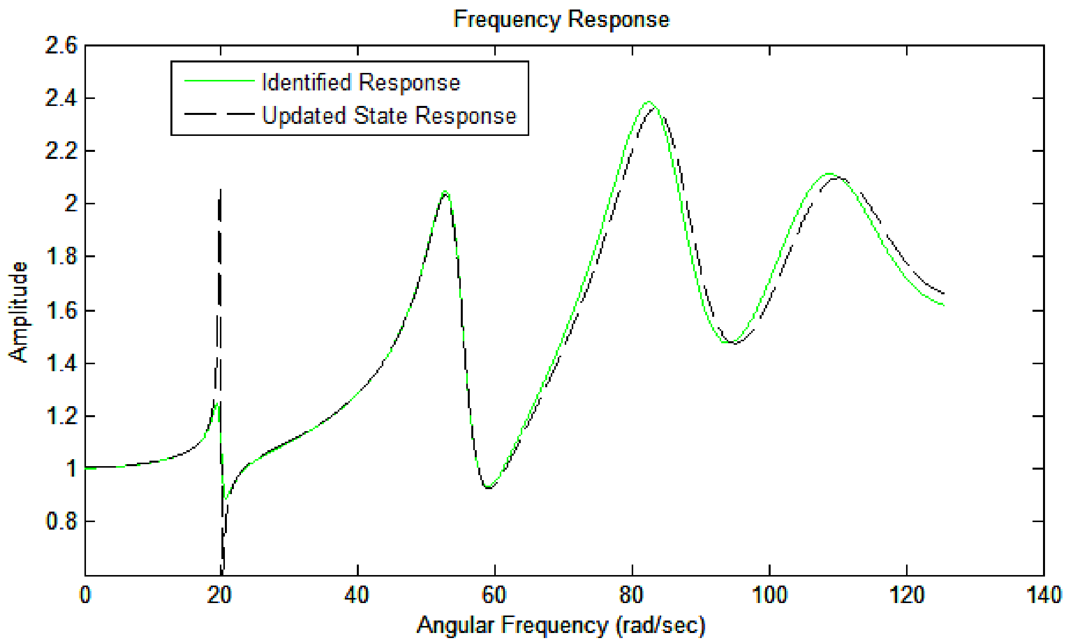

Figure 9: Frequency Responses Using Quadratic Pencil Method (Partial Identification)

The aspect of this method that is most advantageous is that the stiffness and damping matrices are updated directly, meaning that there is a better physical representation of the updated system.

3.5 Constrained Eigenstructure Assignment Method

The constrained eigen-structure assignment method (ESA) is mathematically similar to the state-space ESA method presented by Andy and Shapiro (1983). However, for civil structures, it is often advantageous to be able to update the mass, stiffness, and damping matrices instead of a state matrix; Inman (1994) presented this method in order to accommodate that need. Since the ESA method was created in a state space form, there were several mathematical challenges involved in transitioning the ESA method to update the mass, stiffness and damping matrices; these challenges were solved by including additional steps and introducing an optimization component to the method.

It is assumed that the dynamic equation found below contains matrices **M**, **C**, **K**, which are square with dimensions of *n* x *n*, symmetric, and often banded.

$$\mathbf{M\ddot{x}} + \mathbf{D\dot{x}} + \mathbf{Kx} = \mathbf{0}. \tag{3.40}$$

The vector **x**, and its derivatives have dimensions of *n* x *n,* and represent the states (displacement and velocity) respectively.

The eigenvalue problem can be written as;

$$\left(\mathbf{M}\lambda_i^2 + \mathbf{D}\lambda_i + \mathbf{K}\right)\mathbf{f_i} = \mathbf{0}, \quad i = 1,\ldots,2n \tag{3.41}$$

where the eigenvalues, λ, are complex and the vector, $\mathbf{f}_i$, are the corresponding complex eigenvectors.

Assuming modal tests are performed to measure the eigenvalues and eigenvectors, represented by:

$$\mathbf{\Lambda} = \text{diag}\left[\lambda'_1, \lambda'_2, \lambda'_3, \ldots, \lambda'_{2n}\right], \text{ and } \mathbf{f_i} \tag{3.42}$$

where λ'_i, i = 1,2,3,...s, and $\mathbf{f}'_i$, i = 1,2,3,...,s, are the measured eigenvalues and eigenvectors and i = s+1,...,n are the analytical eigenvalues and eigenvectors that have not been measured and hence, not updated.

In the case of equation (3.40), it is assumed that the updated damping and stiffness matrices are given by:

$$\mathbf{D}' = \mathbf{D} + \overline{\mathbf{D}}, \quad \mathbf{K}' = \mathbf{K} + \overline{\mathbf{K}} \tag{3.43}$$

where $\overline{\mathbf{K}}$ and $\overline{\mathbf{D}}$ represent the updates to the matrices that will be calculated (Inman, 1994) using the ensuing procedure. These are combined to create the feedback gain matrix, **K**, used to assign the measured eigenvalues and eigenvectors. Substituting the updated damping and stiffness matrices in equation (3.43) into equation (3.41) and replacing the analytical with the measured eigenstructure we get

$$\left(\mathbf{M}\lambda'^2_i + \mathbf{D}\lambda'_i + \mathbf{K}\right)\mathbf{f'_i} + \left(\overline{\mathbf{D}}\lambda'_i + \overline{\mathbf{K}}\right)\mathbf{f'_i} = \mathbf{0}. \tag{3.44}$$

which can be rewritten as

$$\left[\left(\mathbf{M}\lambda'^2_i + \mathbf{D}\lambda'_i + \mathbf{K}\right) \mid \mathbf{I}\right]\left[\frac{\mathbf{f'_i}}{\left(\overline{\mathbf{D}}\lambda'_i + \overline{\mathbf{K}}\right)\mathbf{f'_i}}\right] = \mathbf{0}, \quad \text{or} \quad \mathbf{\Gamma_i \Psi_i} = \mathbf{0} \tag{3.45}$$

where $\mathbf{\Gamma}_i$ is and n x 2n matrix and $\mathbf{\Psi}_i$ is a *2n* x *1* matrix, which can be defined as

$$\mathbf{\Psi_i} = \left[\frac{\mathbf{V_i}}{\overline{\mathbf{V}}_\mathbf{i}}\right]\mathbf{e_i} \tag{3.46}$$

where $\left[\frac{\mathbf{V_i}}{\overline{\mathbf{V}}_\mathbf{i}}\right]$ is a *2n* x *n* matrix which contains the orthonormal basis for the null space of the matrix $\mathbf{\Gamma}_i$ and $\mathbf{e}_i$ is an *n* x *1* vector that contains complex coefficients. Then, using equations (3.45) and (3.46) the following can be formed

$$\mathbf{f'_i} = \mathbf{V_i e_i} \quad \text{and} \quad \mathbf{e_i} = \mathbf{V_i^{-1} f'_i} \tag{3.47}$$

However, if the eigenvectors are only partially placed, only the rows of $\mathbf{V}_i$ that correspond to the known values of $\mathbf{f}_i'$ are retained, and the other vectors are temporarily removed, changing the equation to

$$\mathbf{e_i} = \mathbf{V_i^+ f'_i} \tag{3.48}$$

where $\boldsymbol{V}_i^{\dagger}$ is the pseudo inverse of the remaining rows. This is different than what is done in state space, since it is not necessary to reduce the size of the analytical model to accommodate partial placement of the eigenvectors. However, the undefined values of the eigenvectors will not be able to be controlled, which means that they will be changed to any value that satisfies the system. Now, using the denominators of equations (3.45) and (3.46) it can be shown that

$$\left(\overline{\mathbf{D}}\lambda'_i + \overline{\mathbf{K}}\right)\mathbf{f'_i} = \overline{\mathbf{V}}_{\mathbf{i}}\mathbf{e_i} \tag{3.49}$$

And by substituting for $\mathbf{f}'_i$ from equation (3.48):

$$\left(\overline{\mathbf{D}}\lambda'_i + \overline{\mathbf{K}}\right)\mathbf{V}_i\mathbf{e}_i = \overline{\mathbf{V}}_i\mathbf{e}_i \tag{3.50}$$

The damping and stiffness update matrices are the feedback gain matrices for placing only the measured eigenstructure. Equation (3.50) can be written in matrix form for *s* modes as:

$$\overline{\mathbf{D}}\mathbf{V}\mathbf{E}\mathbf{\Lambda}' + \overline{\mathbf{K}}\mathbf{V}\mathbf{E} = \overline{\mathbf{V}}\mathbf{E}. \tag{3.51}$$

Equation (3.52) can be re-written as

$$\begin{bmatrix}\overline{\mathbf{D}} & \overline{\mathbf{K}}\end{bmatrix}\begin{bmatrix}\mathbf{VE\Lambda'} \\ \mathbf{VE}\end{bmatrix} = \overline{\mathbf{V}}\mathbf{E}. \tag{3.52}$$

For the case of complex values in equation (3.52), the real and imaginary parts are separated into

$$\left[\overline{\mathbf{D}} \quad \overline{\mathbf{K}}\right]\begin{bmatrix} \left((\mathbf{VE})_{\mathbf{R}}\boldsymbol{\Lambda}'_{\mathbf{R}} - (\mathbf{VE})_{\mathbf{I}}\boldsymbol{\Lambda}'_{\mathbf{I}}\right) & \left((\mathbf{VE})_{\mathbf{R}}\boldsymbol{\Lambda}'_{\mathbf{I}} - (\mathbf{VE})_{\mathbf{I}}\boldsymbol{\Lambda}'_{\mathbf{R}}\right) \\ (\mathbf{VE})_{\mathbf{R}} & (\mathbf{VE})_{\mathbf{I}} \end{bmatrix} = \left[(\overline{\mathbf{V}}\mathbf{E})_{\mathbf{R}} \quad (\overline{\mathbf{V}}\mathbf{E})_{\mathbf{I}}\right] \tag{3.53}$$

where the subscripts R and I correspond to the real and imaginary parts of the equation. Re-writing the equation:

$$\left[\overline{\mathbf{D}} \quad \overline{\mathbf{K}}\right]\mathbf{G} = \mathbf{H} \tag{3.54}$$

where **G** and **H** represent the matrices in equation (3.53). Note that $\overline{\mathbf{K}}$, $\overline{\mathbf{D}}$, **G**, and **H** are all real valued matrices.

If there are no constraints on the update matrices $\overline{\mathbf{K}}$ and $\overline{\mathbf{D}}$, this equation can be solved by taking the pseudo-inverse of **G** on both sides of the equation (3.54). However, in order to ensure that $\overline{\mathbf{K}}$ and $\overline{\mathbf{D}}$ are both symmetrical and contain the same connectivity as the original damping and stiffness matrices, taking the pseudo-inverse of **G** is not a realistic solution. In order to solve this equation an optimization problem is formulated (Datta, 2000).

The optimization problem is created by finding the minimum of:

$$J = \left\| \left[\overline{\mathbf{D}} \quad \overline{\mathbf{K}}\right]\mathbf{G} - \mathbf{H} \right\|^2 \tag{3.55}$$

This then becomes a problem of non-linear optimization where standard tools available in MATLAB can be utilized. For this document the optimization toolbox in MATLAB was used. Although constraints may be helpful, the only required restraints are upper and lower bounds on each parameter. The parameters are chosen by the user, however, are usually restricted to the non-zero values in the damping and stiffness matrices.

In order to create a usable cost function to input into MATLAB, the gain matrices are written as

$$\mathbf{C} = \begin{bmatrix} \overline{\mathbf{D}} & \overline{\mathbf{K}} \end{bmatrix}. \tag{3.56}$$

The cost function in equation (3.55) can be written as

$$J = \text{trace}\left[(\mathbf{CG} - \mathbf{H})^{\mathbf{T}} (\mathbf{CG} - \mathbf{H}) \right] \tag{3.57}$$

and, derivatives of the cost function with respect to the parameters can be written as

$$\frac{\partial J}{\partial \theta} = \text{trace}\left[\frac{\partial \left((\mathbf{CG} - \mathbf{H})^{\mathbf{T}} (\mathbf{CG} - \mathbf{H}) \right)}{\partial \theta} \right] \tag{3.58}$$

where θ is the parameter that is being evaluated.

This equation can be re-written as

$$\frac{\partial J}{\partial \theta} = 2\,\text{trace}\left(\mathbf{STG} \right) \tag{3.59}$$

where **S** $(\mathbf{CG} - \mathbf{H})^{\mathrm{T}}$

T is the derivative of **C** with respect to the parameter $\boldsymbol{\theta}$.

Using the *fmincon* function in MATLAB along with the equations above, the updating of the damping and stiffness matrix is accomplished.

The MATLAB script for this method can be found in Appendix B6.

Figure 10 displays the frequency responses of the updated and identified states and Table 8 shows the updated and desired eigenvalues. The updated eigenvalues produced using the constrained ESA method vary more than the previous methods, however, their absolute values (or natural frequencies) are very similar to the desired values.

Table 8: Constrained ESA Results

Eigenvalues (desired)	Eigenvalues (updated)
-0.78 - 19.88i	-0.79 - 19.88i
-0.78 + 19.88i	-0.79 + 19.88i
-2.95 - 54.19i	-3.03 - 53.60i
-2.95 + 54.19i	-3.03 + 53.60i
-7.15 - 84.30i	-7.27 - 82.03i
-7.15 + 84.30i	-7.27 + 82.03i
-11.99 - 108.68i	-13.78 - 102.05i
-11.99 + 108.68i	-13.78 + 102.05i
-16.86 - 129.07i	-18.85 - 120.29i
-16.86 + 129.07i	-18.85 + 120.29i

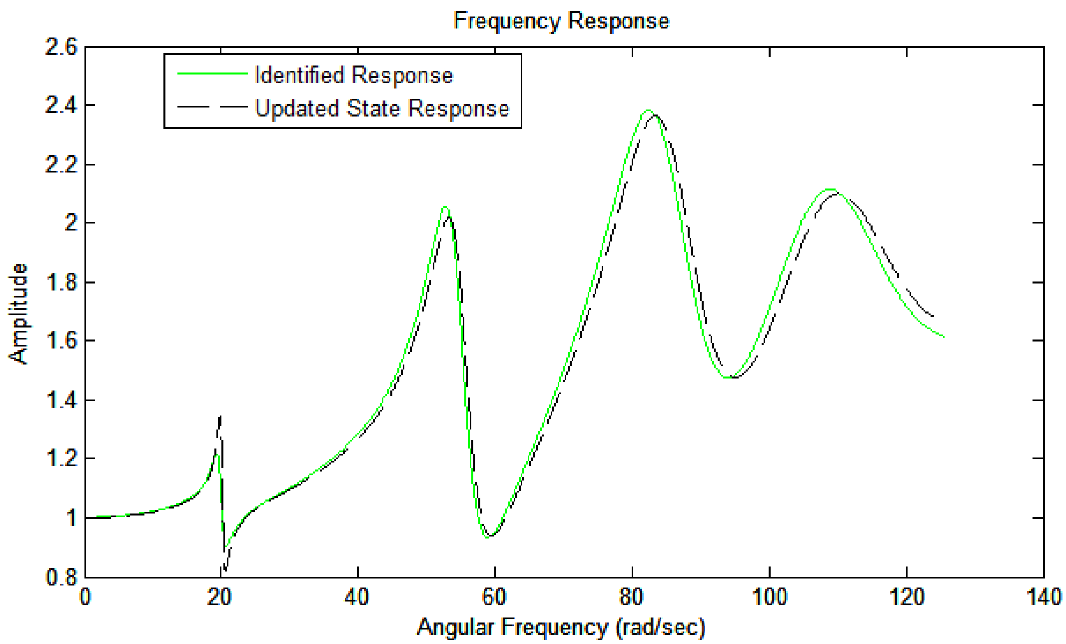

Figure 10: Frequency Responses Using Constrained ESA

The constrained ESA method provides a frequency response that almost exactly replicates that of the identified state.

Figure 11 displays the frequency response and Table 9 shows the updated and desired eigenvalues for the partially identified system. The frequency response is very similar for only two of the modes, whereas it differs significantly for three of them.

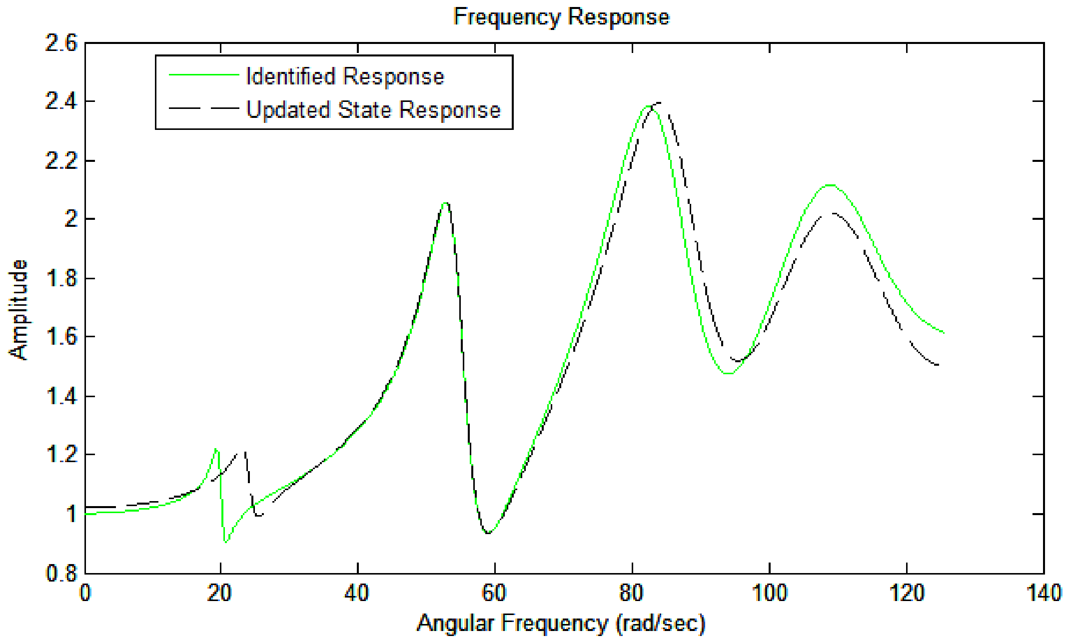

Figure 11: Frequency Responses Using Constrained ESA (Partial Identification)

The updated eigenvalues are significantly different from the desired ones in this case; however, their absolute values are again similar.

Table 9: Constrained ESA Results for Partial Identification

Eigenvalues (desired)	Eigenvalues (updated)
-0.78 - 19.88i	-0.79 - 19.88i
-0.78 + 19.88i	-0.79 + 19.88i
-2.95 - 54.19i	-3.18 - 51.97i
-2.95 + 54.19i	-3.18 + 51.97i
-7.15 - 84.30i	-7.45 - 80.81i
-7.15 + 84.30i	-7.45 + 80.81i
-11.99 - 108.68i	-12.63 - 104.93i
-11.99 + 108.68i	-12.63 + 104.93i
-16.86 - 129.07i	-19.65 - 125.07i
-16.86 + 129.07i	-19.65 + 125.07i

This method is advantageous because it updates both the mass and damping matrices, and can retain the form selected by the user.

3.6 Altered Constrained Eigenstructure Assignment Method

Although the Constrained Eigenstructure Assignment method described by Schulz and Inman (2000) does work, there are times when updating the damping matrix is not what is necessarily desired. Another option is to update the mass matrix instead of the damping matrix. This method is a slight variation of the method described by Schulz and Inman (1994). In this case, the updated matrices are:

$$\mathbf{M'} = \mathbf{M} + \overline{\mathbf{M}}, \quad \mathbf{K'} = \mathbf{K} + \overline{\mathbf{K}} \tag{3.60}$$

and equation (3.45) is:

$$\left[\left(\mathbf{M}\lambda'^2_i + \mathbf{D}\lambda_i + \mathbf{K}\right) \mid \mathbf{I}\right]\left[\frac{\mathbf{f'_i}}{\left(\overline{\mathbf{M}}\lambda'^2_i + \overline{\mathbf{K}}\right)\mathbf{f'_i}}\right] = \mathbf{0} \tag{3.61}$$

which results in:

$$\left[\overline{\mathbf{M}} \quad \overline{\mathbf{K}}\right]\begin{bmatrix} \left((\mathbf{VE})_\mathbf{R}\,\mathbf{\Lambda'^2_R} - (\mathbf{VE})_\mathbf{I}\,\mathbf{\Lambda'^2_I}\right) & \left((\mathbf{VE})_\mathbf{R}\,\mathbf{\Lambda'^2_R} - (\mathbf{VE})_\mathbf{I}\,\mathbf{\Lambda'^2_R}\right) \\ (\mathbf{VE})_\mathbf{R} & (\mathbf{VE})_\mathbf{I} \end{bmatrix} = \left[\left(\overline{\mathbf{VE}}\right)_\mathbf{R} \quad \left(\overline{\mathbf{VE}}\right)_\mathbf{I}\right] \tag{3.62}$$

The only difference in between the Altered Constrained Eigenstructure Assignment method and the Constrained Eigenstructure Assignment method is the use of the mass matrix for the purpose of updating in equation (3.53)

The MATLAB script for this method can be found in B7.

Figure 12 displays the frequency responses of the updated and identified states.

It is apparent that the altered constrained ESA method has provided the closest match to the identified state frequency response.

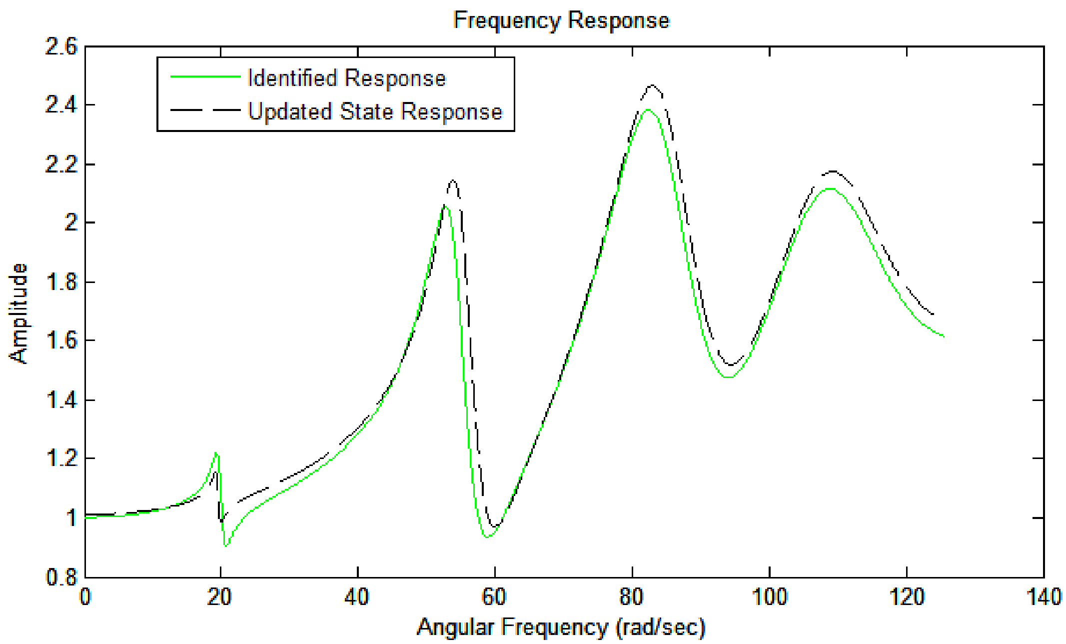

Figure 12: Frequency Responses Using Altered Constrained ESA

Table 10 shows the updated and desired eigenvalues using the altered constrained ESA method. Similar to the constrained ESA method, the updated eigenvalues differ from the desired ones, but their absolute values are similar. However, this method generally provides eigenvalues with a smaller error.

Table 10: Altered Constrained ESA Results

Eigenvalues (desired)	Eigenvalues (updated)
-0.78 - 19.88i	-0.81 - 19.48i
-0.78 + 19.88i	-0.81 + 19.48i
-2.95 - 54.19i	-2.91 - 55.29i
-2.95 + 54.19i	-2.91 + 55.29i
-7.15 - 84.30i	-7.11 - 84.84i
-7.15 + 84.30i	-7.11 + 84.84i
-11.99 - 108.68i	-11.96 - 109.01i
-11.99 + 108.68i	-11.96 + 109.01i
-16.86 - 129.07i	-17.04 - 128.97i
-16.86 + 129.07i	-17.04 + 128.97i

Figure 13 displays the frequency response for the altered constrained ESA method when using partial identification. It is seen that the updated state response is very similar to the identified one.

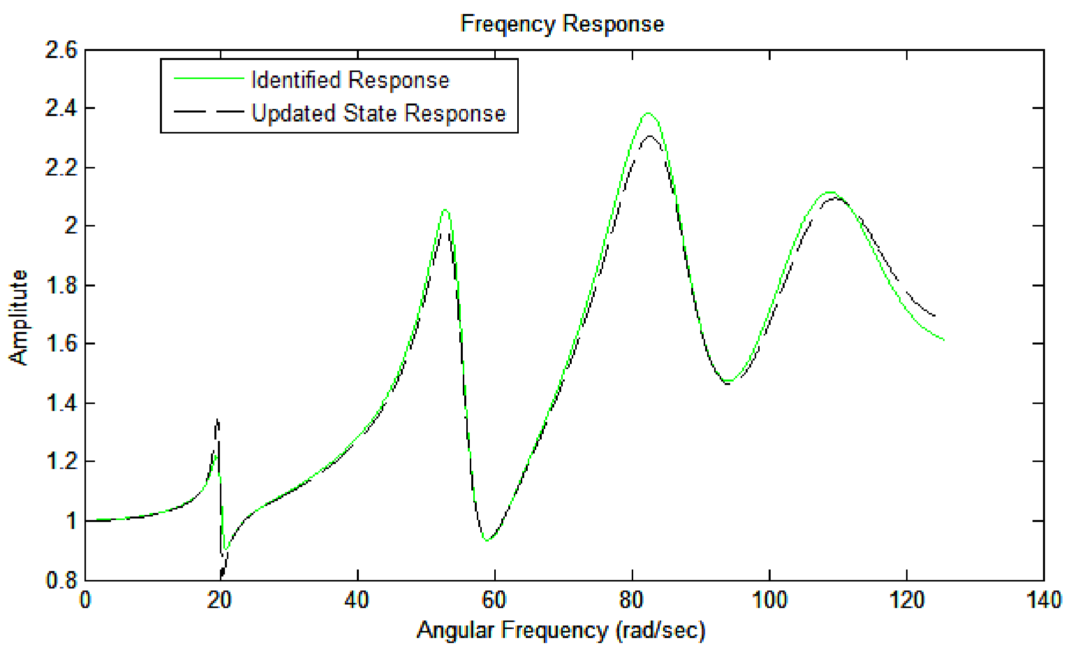

Figure 13: Frequency Responses Using Altered Constrained ESA (Partial Identification)

Table 11 shows the updated and desired eigenvalues. Again, it is seen that the updated eigenvalues vary slightly from their desired counterparts, but their absolute values are very similar. It is observed that the identified eigenvalues (first, second, fifth, and sixth) are closer to the desired ones, whereas the remaining eigenvalues stray further away.

Table 11: Altered Constrained ESA Results for Partial Identification

Eigenvalues (desired)	Eigenvalues (updated)
-0.78 - 19.88i	-0.80 - 19.74i
-0.78 + 19.88i	-0.80 + 19.74i
-2.95 - 54.19i	-3.01 - 54.20i
-2.95 + 54.19i	-3.01 + 54.20i
-7.15 - 84.30i	-7.33 - 84.55i
-7.15 + 84.30i	-7.33 + 84.55i
-11.99 - 108.68i	-12.28 - 109.46i
-11.99 + 108.68i	-12.28 + 109.46i
-16.86 - 129.07i	-17.31 - 129.73i
-16.86 + 129.07i	-17.31 + 129.73i

This method is advantageous because it updates both the mass and stiffness matrices, and can retain whichever form the user chooses. However, this may cause difficulties with larger systems since the choice of terms to be updated affect the optimization procedure greatly.

3.7 Summary

Using the example presented in this section for the control-based model updating methods some general conclusions can be drawn. The main problem is that although the desired eigenvectors can, in theory, be arbitrarily assigned, the value of the real and imaginary components of the desired eigenvalues must be similar to those of the analytical eigenvalues. This is a limitation of all the eigen-structure assignment methods studied in this chapter with the exception of the state space method.

4.0 Full Scale Model

The methods described in the previous chapter are tested on a larger real-life model of a structure. For this purpose, a FE model was developed using the structural drawings and a commercially available program, SAP2000©, and the eigenstructure extracted from it.

The mass properties of the full scale model were estimated, using the structural and architectural drawings, which then formed the inputs into SAP2000© and are, therefore, known. The eigenvalues and eigenvector matrices were calculated by SAP2000©. First, an orthogonality check was done to ensure that equation (4.1) was approximately equal to an identity matrix, and then the stiffness matrix is calculated according to

$$\mathbf{\Phi}^{\mathbf{T}}\mathbf{M}\mathbf{\Phi} \tag{4.1}$$

$$\mathbf{K} = \left(\mathbf{\Phi}^{\mathbf{T}}\right)^{-1}\mathbf{\Lambda}\mathbf{\Phi}^{-1} \tag{4.2}$$

Where $\mathbf{\Phi}$ is the SAP2000© calculated eigenvector matrix

$\mathbf{\Lambda}$ is the SAP2000© calculated eigenvalue matrix, and

$\mathbf{M}$ is the estimated mass matrix.

The damping matrix was then found using equations (4.3) and (4.4).

$$\hat{\mathbf{C}} = 0.05\sqrt{\mathbf{\Lambda}} \tag{4.3}$$

$$\mathbf{C} = \mathbf{M}\mathbf{\Phi}\hat{\mathbf{C}}\mathbf{\Phi}^{\mathbf{T}}\mathbf{M} \tag{4.4}$$

However, this process yields completely filled stiffness and damping matrices, because of this, these matrices lose much of their physical meaning. Figure 14 shows the FE model created using SAP2000©.

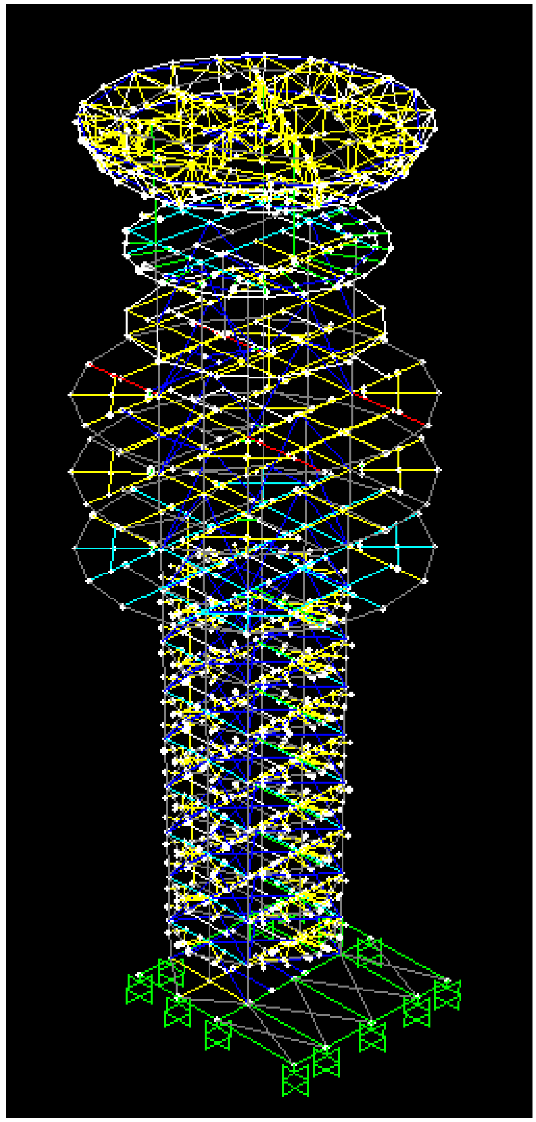

Figure 14: Finite-element Model

4.1 Reduced-Order Full Scale Model

Since the full scale model is large, a reduced-order model is calculated to evaluate each of the model updating method's efficiency and to test the model reduction methods.

The 42 degrees of freedom system that is described earlier is reduced to 18 degrees of freedom using each of the model reduction methods described in section 2, and the results were compared. It can be seen in Table 12 and Table 13 that all of the methods produced results for the first four modes with a good degree of accuracy; however, the Iterated IRS and SEREP methods were both able to re-produce the first 18 degrees of freedom with very little error. Table 14 shows the MAC and MACM values for the 18 modes after using the Iterated IRS and SEREP methods.

Table 12: Results for the Order of the Full Scale Model

	Actual		**Static**			**IRS**		
Mode	***f (Hz)***	**Period (s)**	***f (Hz)***	**Period (s)**	**Error (%)**	***f (Hz)***	**Period (s)**	**Error (%)**
1	0.656	1.523	0.656	1.523	0.01%	0.656	1.523	0.00%
2	0.919	1.088	0.919	1.088	0.04%	0.919	1.088	0.00%
3	1.402	0.713	1.403	0.713	0.01%	1.402	0.713	0.00%
4	2.734	0.366	2.738	0.365	0.16%	2.734	0.366	0.00%
5	3.013	0.332	3.031	0.330	0.60%	3.013	0.332	0.00%
6	3.363	0.297	3.398	0.294	1.06%	3.363	0.297	0.01%
7	4.594	0.218	4.637	0.216	0.95%	4.594	0.218	0.02%
8	6.083	0.164	6.699	0.149	10.12%	6.184	0.162	1.66%
9	6.710	0.149	7.238	0.138	7.87%	6.742	0.148	0.48%
10	9.010	0.111	10.014	0.100	11.14%	9.430	0.106	4.66%
11	10.090	0.099	11.510	0.087	14.07%	10.655	0.094	5.60%
12	11.199	0.089	12.110	0.083	8.14%	12.040	0.083	7.51%
13	12.022	0.083	14.466	0.069	20.33%	13.084	0.076	8.83%
14	12.027	0.083	16.098	0.062	33.84%	14.407	0.069	19.78%
15	13.834	0.072	18.898	0.053	36.61%	17.927	0.056	29.59%
16	14.409	0.069	21.755	0.046	50.99%	19.993	0.050	38.76%
17	15.719	0.064	24.680	0.041	57.01%	23.534	0.042	49.72%
18	16.838	0.059	30.910	0.032	83.57%	30.582	0.033	81.62%

Table 13: Results for Reducing the Order of the Full Scale Model

	Actual		Iterated IRS			SEREP		
Mode	***f (Hz)***	**Period (s)**	***f (Hz)***	**Period (s)**	**Error (%)**	***f (Hz)***	**Period (s)**	**Error (%)**
1	0.656	1.523	0.656	1.523	0.00%	0.656	1.523	0.00%
2	0.919	1.088	0.919	1.088	0.00%	0.919	1.088	0.00%
3	1.402	0.713	1.402	0.713	0.00%	1.402	0.713	0.00%
4	2.734	0.366	2.734	0.366	0.00%	2.734	0.366	0.00%
5	3.013	0.332	3.013	0.332	0.00%	3.013	0.332	0.00%
6	3.363	0.297	3.363	0.297	0.00%	3.363	0.297	0.00%
7	4.594	0.218	4.594	0.218	0.00%	4.593	0.218	0.00%
8	6.083	0.164	6.083	0.164	0.00%	6.083	0.164	0.00%
9	6.710	0.149	6.710	0.149	0.00%	6.710	0.149	0.00%
10	9.010	0.111	9.010	0.111	0.00%	9.010	0.111	0.00%
11	10.090	0.099	10.090	0.099	0.00%	10.090	0.099	0.00%
12	11.199	0.089	11.199	0.089	0.00%	11.199	0.089	0.00%
13	12.022	0.083	12.022	0.083	0.00%	12.022	0.083	0.00%
14	12.027	0.083	12.027	0.083	0.00%	12.027	0.083	0.00%
15	13.834	0.072	13.834	0.072	0.00%	13.834	0.072	0.00%
16	14.409	0.069	14.409	0.069	0.00%	14.409	0.069	0.00%
17	15.719	0.064	15.719	0.064	0.00%	15.719	0.064	0.00%
18	16.838	0.059	16.838	0.059	0.00%	16.838	0.059	0.00%

Table 14: MAC and MAC Values for Iterated IRS and SEREP Reduction of Full Scale Model

	SEREP		Iterated IRS	
Mode	**MAC**	**MACM**	**MAC**	**MACM**
1	1.00	1.00	1.00	1.00
2	1.00	1.00	1.00	1.00
3	1.00	1.00	1.00	1.00
4	1.00	1.00	1.00	1.00
5	1.00	1.00	1.00	1.00
6	1.00	1.00	1.00	1.00
7	1.00	1.00	1.00	1.00
8	1.00	1.00	1.00	1.00
9	1.00	1.00	1.00	1.00
10	1.00	1.00	1.00	1.00
11	1.00	1.00	1.00	1.00
12	1.00	1.00	1.00	1.00
13	0.99	0.87	0.99	0.87
14	0.73	0.87	0.72	0.88
15	1.00	1.00	1.00	1.00
16	1.00	1.00	1.00	1.00
17	1.00	1.00	1.00	1.00
18	1.00	1.00	1.00	0.95

4.2 Lagrange Multiplier (Stiffness Matrix Updated)

Equation (4.1) confirms that the eigenvectors are othogonalized with the mass matrix; therefore, it is not necessary to update the measure eigenvectors using equation (2.14). Equation (3.11) is used to update the stiffness matrix, and then the resulting modal parameters are calculated using equations (1.4) and (1.5). Table 15 displays the first 18 modes of the system, the analytical eigenvalues, the measured eigenvalues; the eigenvalues calculated using the updated matrices, the error between the measured and updated eigenvalues, and the MAC values for the corresponding eigenvectors.

Table 15: Results for Direct Stiffness Model Updating on Full Scale Model

Modes	Analytical Eigenvalues	Measured Eigenvalues	Updated System Eigenvalues	Percent Error (%)	MAC
1	17.01	17.55	17.55	0.00	1.00
2	33.33	34.08	34.08	0.00	1.00
3	77.79	78.13	78.13	0.00	1.00
4	294.93	295.49	295.49	0.00	1.00
5	358.36	359.33	359.33	0.00	1.00
6	446.49	447.02	447.02	0.00	1.00
7	832.67	833.97	833.97	0.00	1.00
8	1460.90	1462.18	1462.18	0.00	1.00
9	1779.90	1778.80	1778.80	0.00	1.00
10	3205.60	3205.64	3205.64	0.00	1.00
11	4018.80	4021.76	4021.76	0.00	1.00
12	4951.80	4952.40	4952.40	0.00	1.00
13	5705.00	5704.82	5704.82	0.00	1.00
14	5705.00	5713.55	5713.55	0.00	1.00
15	7551.10	7559.01	7559.01	0.00	1.00
16	8200.40	8197.37	8197.37	0.00	1.00
17	9759.80	9756.42	9756.42	0.00	1.00
18	11179.00	11194.39	11194.39	0.00	1.00

With limited transducers, it is only possible to estimate the lower modes. For this situation, it is assumed that the first 18 modes can be determined. Appendix C1 contains the complete set of results.

However, the stiffness matrix has larger values away from the diagonals than in its original form. So, even though the eigenvalues and eigenvectors are updated almost perfectly, the system is now further away from a physical representation of the actual structure.

It should be noted though, that for this specific use, the stiffness matrix is already completely filled and since this method does not update the mass matrix, it is still diagonal.

The reduced order full scale model is also updated using this technique. The resulting modal parameters for the first 18 modes are listed in Table 16. The results are nearly exactly the same as the ones produced for the full 42 degree of freedom model (Table 15); this is to be expected, since the direct method and the SEREP reduction method are both mathematical methods that do not take physical representation into account.

Table 16: Results for Direct Stiffness Model Updating on Order-Reduced Full Scale Model

Modes	Analytical Eigenvalues	Measured Eigenvalues	Updated System Eigenvalues	Percent Error (%)	MAC
1	17.01	17.55	17.55	0.00	1.00
2	33.33	34.08	34.08	0.00	1.00
3	77.79	78.13	78.13	0.00	1.00
4	294.93	295.49	295.49	0.00	1.00
5	358.36	359.33	359.32	0.00	1.00
6	446.49	447.02	447.03	0.00	1.00
7	832.67	833.97	833.94	0.00	1.00
8	1460.90	1462.18	1462.18	0.00	1.00
9	1779.90	1778.80	1778.79	0.00	1.00
10	3205.60	3205.64	3205.62	0.00	1.00
11	4018.80	4021.76	4021.76	0.00	1.00
12	4951.80	4952.40	4952.41	0.00	1.00
13	5705.00	5704.82	5704.84	0.00	1.00
14	5705.00	5713.55	5713.55	0.00	1.00
15	7551.10	7559.01	7559.01	0.00	1.00
16	8200.40	8197.37	8197.37	0.00	1.00
17	9759.80	9756.42	9756.42	0.00	1.00
18	11179.00	11194.39	11194.39	0.00	1.00

However, since the mass matrix is now full from the SEREP reduction, it would be beneficial to have the ability to update it as well. It is noted that the stiffness matrix has very large changes to some of its values; this is likely because of the fact that the mass matrix is full and cannot be updated.

4.3 Lagrange Multiplier (Stiffness and Mass Matrices Updated)

Using the raw measurement direct model updating method, the mass and stiffness matrices were updated with equations (3.31) and (3.32). The results for the first 18 modes are presented in Table 17. The results are similar to the direct method (stiffness), which is to be expected, since they are both based on similar optimization procedures, the only difference is in the matrices being updated. However, it is noted that the mass matrix is no longer diagonal; since the stiffness matrix is already not a physical representation it is more beneficial to update only the stiffness matrix. The full 42 degree of freedom results are found in Appendix C2.

The raw measurement direct model updating method was used to update the order-reduced full scale model as well. For this situation, both the mass and stiffness matrices were filled, so the fact that the mass matrix was updated does not matter. In fact, because the mass matrix was able to be updated, the updated stiffness and mass matrices are closer to their original form. The results, in Table 17, show that they eigensystem was reproduced exactly.

Table 17: Results for Raw Measurement Direct Model Updating on Full Scale Model

Modes	Analytical Eigenvalues	Measured Eigenvalues	Updated System Eigenvalues	Percent Error (%)	MAC
1	17.01	17.55	17.55	0.00	1.00
2	33.33	34.08	34.08	0.00	1.00
3	77.79	78.13	78.13	0.00	1.00
4	294.93	295.49	295.49	0.00	1.00
5	358.36	359.33	359.33	0.00	1.00
6	446.49	447.02	447.02	0.00	1.00
7	832.67	833.97	833.97	0.00	1.00
8	1460.90	1462.18	1462.18	0.00	1.00
9	1779.90	1778.80	1778.80	0.00	1.00
10	3205.60	3205.64	3205.64	0.00	1.00
11	4018.80	4021.76	4021.76	0.00	1.00
12	4951.80	4952.40	4952.40	0.00	1.00
13	5705.00	5704.82	5704.82	0.00	1.00
14	5705.00	5713.55	5713.55	0.00	1.00
15	7551.10	7559.01	7559.01	0.00	1.00
16	8200.40	8197.37	8197.37	0.00	1.00
17	9759.80	9756.42	9756.42	0.00	1.00
18	11179.00	11194.39	11194.39	0.00	1.00

Table 18: Results for Raw Measurement Direct Model Updating on Order-Reduced Full Scale Model

Modes	Analytical Eigenvalues	Measured Eigenvalues	Updated System Eigenvalues	Percent Error (%)	MAC
1	17.01	17.55	17.55	0.00	1.00
2	33.33	34.08	34.08	0.00	1.00
3	77.79	78.13	78.13	0.00	1.00
4	294.93	295.49	295.49	0.00	1.00
5	358.36	359.33	359.32	0.00	1.00
6	446.49	447.02	447.02	0.00	1.00
7	832.67	833.97	833.95	0.00	1.00
8	1460.90	1462.18	1462.18	0.00	1.00
9	1779.90	1778.80	1778.79	0.00	1.00
10	3205.60	3205.64	3205.64	0.00	1.00
11	4018.80	4021.76	4021.75	0.00	1.00
12	4951.80	4952.40	4952.40	0.00	1.00
13	5705.00	5704.82	5704.79	0.00	1.00
14	5705.00	5713.55	5713.51	0.00	1.00
15	7551.10	7559.01	7559.01	0.00	1.00
16	8200.40	8197.37	8197.37	0.00	1.00
17	9759.80	9756.42	9756.42	0.00	1.00
18	11179.00	11194.39	11194.39	0.00	1.00

4.4 Penalty Function Method

The penalty function method for a 42 degree of freedom system was computationally difficult; however, the fact that the stiffness matrix is full also adds a component of judgement in the user. For this example, 315 parameters were chosen, and each was weighted according to judgement. The MATLAB script used can be found in Appendix C3. It should be noted that because of the very large eigenvalues for the larger modes, only the first 18 were taken as measured eigenvalues. The reason for this is that the ratio of the largest eigenvalue to the smallest is over 10 000, and since the smaller eigenvalues are more important, they received a higher weighting. If the appropriate weightings were used, most of the eigenvalues after the 18th mode would be weighted near 0 anyway.

The main advantage of this method is that the parameters can be chosen by the user according to the situation, and the weightings are easily applied. Because the parameters can be selected, this method is capable of updating the mass matrix without losing its diagonal property.

The error (in percentage) between the measured and updated eigenvalues is more than the direct methods; however, they are relatively small and the MAC values for all 42 modes were 1.00. The full results for this method can be found in Appendix C4, and the results for the first 18 modes are found in Table 19.

Table 19: Results for the Penalty Function Model Updating Method on Full Scale Model

Modes	Analytical Eigenvalues	Measured Eigenvalues	Updated System Eigenvalues	Percent Error (%)	MAC
1	17.01	17.55	17.59	0.25	1.00
2	33.33	34.08	33.90	0.54	1.00
3	77.79	78.13	78.11	0.03	1.00
4	294.93	295.49	295.68	0.07	1.00
5	358.36	359.33	359.59	0.07	1.00
6	446.49	447.02	447.00	0.01	1.00
7	832.67	833.97	834.03	0.01	1.00
8	1460.90	1462.18	1461.74	0.03	1.00
9	1779.90	1778.80	1777.65	0.07	1.00
10	3205.60	3205.64	3204.98	0.02	1.00
11	4018.80	4021.76	4020.95	0.02	1.00
12	4951.80	4952.40	4951.81	0.01	1.00
13	5705.00	5704.82	5704.04	0.01	1.00
14	5705.00	5713.55	5711.10	0.04	1.00
15	7551.10	7559.01	7556.68	0.03	1.00
16	8200.40	8197.37	8196.81	0.01	1.00
17	9759.80	9756.42	9756.12	0.00	1.00
18	11179.00	11194.39	11193.94	0.00	1.00

The reduced-order full scale model created some problems with the penalty function method due to the difficulty in choosing updating parameters to be updated and their corresponding weights. As well, this caused the larger eigenvalues to have more of an effect, so only the first 3 modes were chosen as measured values. The results are found in Table 20. The issues caused by the reduced-order full scale model are apparent when observing the percent error between the measured eigenvalues and updated eigenvalues directly following the modes that were chosen to be updated. The 4th, 5th, and 6th modes had errors or 2.26%, 9.80%, and

11.22% respectively. If all 18 modes have measured values, the results found using the penalty method are not acceptable.

Table 20: Results for the Penalty Function Model Updating Method on Reduced-Order Full Scale Model

Modes	Analytical Eigenvalues	Measured Eigenvalues	Updated System Eigenvalues	Percent Error (%)	MAC
1	17.01	17.55	17.55	0.00	1.00
2	33.33	34.08	34.08	0.00	1.00
3	77.79	78.13	78.13	0.00	0.99
4	294.93	295.49	302.18	2.26	0.98
5	358.36	359.33	394.54	9.80	0.99
6	446.49	447.02	497.18	11.22	0.99
7	832.67	833.97	843.57	1.15	0.98
8	1460.90	1462.18	1457.27	0.34	0.98
9	1779.90	1778.80	1774.28	0.25	0.98
10	3205.60	3205.64	3218.98	0.42	1.00
11	4018.80	4021.76	4012.89	0.22	1.00
12	4951.80	4952.40	4854.18	1.98	0.99
13	5705.00	5704.82	5444.13	4.57	1.00
14	5705.00	5713.55	5699.60	0.24	0.91
15	7551.10	7559.01	7550.42	0.11	1.00
16	8200.40	8197.37	8152.57	0.55	1.00
17	9759.80	9756.42	9657.29	1.02	1.00
18	11179.00	11194.39	11183.19	0.10	1.00

In this example, the MAC values for each mode are relatively good, however, when the calculations are performed using the first four modes, the 12^{th} and 13^{th} modes were not represented correctly.

The penalty method, although very useful for a smaller, simpler model, becomes difficult to use and obtain proper results when using a larger, more complicated model.

4.5 Quadratic Pencil Method

One of the main issues with this method is that there is no control over which parameters within the stiffness and damping matrices are updated. It also does not update the mass matrix, which could be desired; however, if it did, the mass matrix would lose its diagonal property. Since this a control based method that uses a direct approach, the calculation of the updated stiffness and damping matrices was simple.

The results for the modal parameters are presented in Table 21. Although there was a small error in some of the updated eigenvalues, this method was able to almost exactly reproduce the measured values. It was able to reproduce the exact eigenvectors for all 42 modes. Also, the stiffness and damping matrices were seen to have very little change, meaning that this method was able to preserve the little physical representation left.

Since the quadratic pencil model updating method worked so well for the full scale model, it was expected that the use of this method on the reduced-ordered full scale model would yield similarly accurate results. However, as it can be seen in Table 22 some of the eigenvalues and eigenvectors had considerable errors. An explanation for this may be that a damping matrix was added to the order-reduced model after the mass and stiffness matrices were reduced, but that alone should not cause the magnitude of error observed from this example.

Table 21: Results for the Quadratic Pencil Model Updating Method on Full Scale Model

Modes	Analytical Eigenvalues	Measured Eigenvalues	Updated System Eigenvalues	Percent Error (%)	MAC
1	17.01	17.55	17.55	0.00	1.00
2	33.33	34.08	34.10	0.03	1.00
3	77.79	78.13	78.15	0.01	1.00
4	294.93	295.49	295.53	0.00	1.00
5	358.36	359.33	359.32	0.00	1.00
6	446.49	447.02	446.98	0.00	1.00
7	832.67	833.97	834.03	0.00	1.00
8	1460.90	1462.18	1462.28	0.00	1.00
9	1779.90	1778.80	1778.70	0.00	1.00
10	3205.60	3205.64	3205.81	0.00	1.00
11	4018.80	4021.76	4021.86	0.00	1.00
12	4951.80	4952.40	4953.09	0.00	1.00
13	5705.00	5704.82	5707.06	0.00	1.00
14	5705.00	5713.55	5710.18	0.00	1.00
15	7551.10	7559.01	7558.89	0.00	1.00
16	8200.40	8197.37	8197.05	0.00	1.00
17	9759.80	9756.42	9756.23	0.00	1.00
18	11179.00	11194.39	11193.71	0.00	1.00

Table 22: Results for the Quadratic Pencil Model Updating Method on Reduced-Order Full Scale Model

Modes	Analytical Eigenvalues	Measured Eigenvalues	Updated System Eigenvalues	Percent Error (%)	MAC
1	17.01	17.55	17.52	0.20	1.00
2	33.33	34.08	33.97	0.32	1.00
3	77.79	78.13	84.73	8.42	0.96
4	294.93	295.49	299.16	1.23	1.00
5	358.36	359.33	359.28	0.01	1.00
6	446.49	447.02	469.62	5.06	1.00
7	832.67	833.97	840.50	0.78	1.00
8	1460.90	1462.18	1667.91	14.06	0.81
9	1779.90	1778.80	1954.66	9.89	0.05
10	3205.60	3205.64	3749.43	16.98	0.92
11	4018.80	4021.76	4035.78	0.35	0.99
12	4951.80	4952.40	5132.51	3.62	0.16
13	5705.00	5704.82	5673.79	0.58	0.05
14	5705.00	5713.55	5717.97	0.14	0.16
15	7551.10	7559.01	5721.63	24.31	0.00
16	8200.40	8197.37	7549.31	7.90	0.06
17	9759.80	9756.42	8226.90	15.68	0.16
18	11179.00	11194.39	10951.71	2.16	0.98

4.6 State Space Method

A major problem arose when using the full state feedback updating method on the control tower, the rank condition of the controllability matrix was not acceptable. The values inside the **B** matrix were changed multiple times; however, none of them were sufficient. In order to address this issue, a controllable form for the state matrices needed to be developed.

For this purpose, the balanced realization of the system was found in MATLAB, which made the system controllable. After this reduction, the state matrix, **A**, was reduced to a 12 x 12 completely filled matrix.

$$\mathbf{A_1} = \begin{bmatrix} -0.042 & 4.086 & 0.001 & 0.001 & -0.019 & -0.001 \\ -4.164 & -0.042 & 0.001 & 0.001 & -0.019 & -0.001 \\ -0.001 & -0.001 & -0.058 & 5.72 & -0.006 & 0.001 \\ -0.001 & -0.001 & -5.83 & -0.059 & -0.006 & 0.001 \\ 0.052 & 0.052 & 0.015 & 0.015 & -0.38 & 18.92 \\ 0.002 & 0.002 & -0.001 & -0.001 & -18.97 & -0.003 \\ 0.004 & 0.004 & -0.071 & -0.072 & -0.15 & 0.00 \\ 0.001 & 0.001 & 0.002 & 0.002 & -0.05 & -0.01 \\ 0.045 & 0.045 & -0.008 & -0.008 & 1.00 & 0.076 \\ 0.001 & 0.001 & -0.007 & -0.008 & -0.097 & -0.007 \\ -0.011 & -0.011 & -0.081 & -0.082 & -0.30 & -0.001 \\ -0.005 & -0.005 & 0.005 & 0.005 & -0.08 & -0.009 \end{bmatrix} \tag{4.5}$$

$$\mathbf{A_2} = \begin{bmatrix} -0.002 & -0.00 & -0.006 & -0.001 & 0.001 & 0.001 \\ -0.002 & -0.00 & -0.006 & -0.001 & 0.001 & 0.001 \\ 0.031 & -0.002 & 0.004 & 0.003 & 0.03 & -0.004 \\ 0.031 & -0.002 & 0.004 & 0.003 & 0.03 & -0.004 \\ 0.12 & 0.018 & -0.81 & 0.041 & 0.28 & 0.051 \\ -0.028 & 0.006 & -0.03 & 0.00 & -0.011 & 0.006 \\ -0.43 & 21.18 & -0.55 & -0.19 & -2.02 & 0.114 \\ -21.19 & -0.004 & -0.016 & 0.02 & 0.039 & -0.016 \\ 0.60 & 0.058 & -0.58 & -28.84 & -1.26 & -0.19 \\ 0.15 & -0.02 & 28.91 & -0.008 & -0.21 & 0.103 \\ -2.06 & 0.023 & 1.43 & 0.10 & -0.77 & 38.10 \\ -0.05 & 0.012 & 0.30 & -0.11 & -38.11 & -0.009 \end{bmatrix} \tag{4.6}$$

where

$$\mathbf{A} = [\mathbf{A_1} \quad \mathbf{A_2}] \tag{4.7}$$

and the **B** matrix is

$$\mathbf{B_1} = \begin{bmatrix} -0.013 & -1.21 & 0 & -0.015 & -1.02 & 0 \\ -0.012 & -1.22 & 0 & -0.014 & -1.03 & 0 \\ 1.164 & -0.016 & 0 & 0.977 & -0.015 & 0 \\ 1.178 & -0.018 & 0 & 0.988 & -0.016 & 0 \\ -0.40 & 2.35 & 0 & -0.043 & 0.133 & 0 \\ 0.017 & 0.205 & 0 & 0.005 & -0.03 & 0 \\ 2.16 & 0.504 & -0.00 & 0.205 & -0.093 & 0 \\ 0.102 & 0.127 & 0 & -0.063 & -0.02 & 0 \\ 0.22 & -1.11 & -0.00 & 0.067 & 1.16 & 0.00 \\ -0.013 & 0.207 & 0 & 0.115 & -0.049 & 0 \\ -1.14 & 0.163 & 0 & 1.66 & -0.345 & 0.00 \\ -0.17 & 0.102 & 0 & 0.03 & -0.135 & 0 \end{bmatrix} \tag{4.8}$$

$$\mathbf{B_2} = \begin{bmatrix} -0.014 & -0.913 & 0 & -0.014 & -0.813 & 0 \\ -0.012 & -0.918 & 0 & -0.014 & -0.817 & 0 \\ 0.887 & -0.016 & 0 & 0.808 & -0.015 & 0 \\ 0.897 & -0.017 & 0 & 0.817 & -0.017 & 0 \\ 0.037 & -0.238 & 0 & 0.089 & -0.493 & 0 \\ 0.002 & -0.063 & 0 & -0.002 & -0.078 & 0 \\ -0.207 & -0.17 & 0 & -0.486 & -0.276 & 0 \\ -0.074 & -0.041 & 0 & -0.057 & -0.06 & 0 \\ 0.003 & 1.37 & 0.00 & -0.053 & 1.33 & 0.00 \\ 0.094 & -0.084 & 0 & 0.057 & -0.095 & 0 \\ 1.51 & -0.36 & 0.00 & 1.06 & -0.101 & 0.00 \\ 0.012 & -0.146 & 0 & 0.022 & -0.117 & 0 \end{bmatrix} \tag{4.9}$$

where

$$\mathbf{B} = \begin{bmatrix} \mathbf{B_1} & \mathbf{B_2} \end{bmatrix} \tag{4.10}$$

Since the model is now reduced to 12 degrees of freedom, there will only be 6 real eigenvalues that will be updated. Also, since the reduction is based solely on the HSVs, the eigenvalues of the system that remain are not necessarily the same as the previous examples. Table 23 contains a summary of the updated eigenvalues.

Table 23: Results for the State Space Model Updating Method on Full Scale Model

Modes	Analytical Eigenvalues	Measured Eigenvalues	Updated System Eigenvalues	Percent Error (%)
1	17.012	16.997	16.997	0.00
2	33.326	33.337	33.337	0.00
3	358.31	358.248	358.248	0.00
4	446.67	446.7898	446.7898	0.00
5	832.89	832.930	832.930	0.00
6	1463.00	1462.992	1462.992	0.00

The MAC values for the eigenvectors were

$$\mathbf{MAC} = \begin{Bmatrix} 0.97 \\ 0.99 \\ 1.00 \\ 0.95 \\ 1.00 \\ 1.00 \end{Bmatrix}. \tag{4.11}$$

The advantage of this method is that the updated eigenvalues and the eigenvectors matched up nearly perfectly. However, the major disadvantages are that the eigenvalues updated are not chosen by the user and may not correspond adequately to the position of the transducers; the state matrix was completely filled and does not resemble a physical representation of the structure; and only 6 eigenvalues and their corresponding eigenvectors were able to be updated.

4.7 Constrained ESA Method

This method is similar to the penalty function method, in that, because the stiffness matrix is filled, there are many options for selecting the parameters. However, this method becomes even more computationally difficult and uses a non-linear optimization function (performed in MATLAB using fmincon). This specific example took 35 minutes to run.

Table 24: Results for the Constrained Eigenstructure Assignment Method Model Updating Method on Full Scale Model

Modes	Analytical Eigenvalues	Measured Eigenvalues	Updated System Eigenvalues	Percent Error (%)	MAC
1	17.01	17.55	17.62	0.39	1.00
2	33.33	34.08	33.20	2.61	1.00
3	77.79	78.13	77.66	0.63	1.00
4	294.93	295.49	295.07	0.15	1.00
5	358.36	359.33	359.11	0.06	1.00
6	446.49	447.02	446.34	0.14	1.00
7	832.67	833.97	833.83	0.02	1.00
8	1460.90	1462.18	1460.84	0.10	1.00
9	1779.90	1778.80	1777.54	0.06	1.00
10	3205.60	3205.64	3204.74	0.03	1.00
11	4018.80	4021.76	4020.35	0.04	1.00
12	4951.80	4952.40	4952.08	0.02	1.00
13	5705.00	5704.82	5705.75	0.02	0.99
14	5705.00	5713.55	5708.93	0.02	0.77
15	7551.10	7559.01	7555.85	0.04	1.00
16	8200.40	8197.37	8195.85	0.02	1.00
17	9759.80	9756.42	9754.98	0.01	1.00
18	11179.00	11194.39	11192.763	0.00	1.00

The results in Table 24 were achieved after only five iterations. When the iteration number was increased to ten, all eigenvalues were identified correctly; however, the MAC value of the 14th mode was decreased to 0.54. The results for all 42 degrees of freedom can be found in Appendix C6.

For the reduced-order model using the Constrained Eigenstructure Assignment method it was difficult to choose parameters to update within the damping and stiffness matrices, since they are both filled. This difficulty is apparent in the quality of updating. Many combinations of parameters were chosen, however, the results were still considerably poor. Table 25 displays the results, and it can be seen that even though some of the eigenvalues and eigenvectors matched up nearly perfectly, many of them did not.

Table 25: Results for Constrained Eigenstructure Assignment Method Model Updating on Order-Reduced Full Scale Model

Modes	Analytical Eigenvalues	Measured Eigenvalues	Updated System Eigenvalues	Percent Error (%)	MAC
1	17.01	17.55	18.03	2.73	0.80
2	33.33	34.08	31.01	9.03	0.56
3	77.79	78.13	76.90	1.61	0.55
4	294.93	295.49	305.70	3.44	0.67
5	358.36	359.33	365.55	1.73	0.99
6	446.49	447.02	432.75	3.18	0.93
7	832.67	833.97	847.77	1.65	0.99
8	1460.90	1462.18	1519.28	3.90	0.95
9	1779.90	1778.80	1699.60	4.45	0.42
10	3205.60	3205.64	3199.30	0.20	1.00
11	4018.80	4021.76	3992.48	0.73	1.00
12	4951.80	4952.40	4929.35	0.48	0.96
13	5705.00	5704.82	5631.11	1.33	1.00
14	5705.00	5713.55	5708.52	0.03	0.84
15	7551.10	7559.01	7556.43	0.03	1.00
16	8200.40	8197.37	8202.92	0.07	0.95
17	9759.80	9756.42	9751.81	0.05	1.00
18	11179.00	11194.39	11045.66	1.32	1.00

4.8 Altered Constrained ESA Method

This method has the same advantages and disadvantages as the Constrained Eigenstructure Assignment method. However, it does have the advantage of updating the mass matrix. Table 26 displays the results of the full scale model update. Again, the first three eigenvalues have a slight error, but that error dissipates as the mode number increases. The main difference between these results and the results from the Constrained Eigenstructure Assignment method is that the MAC values for all 42 modes are 1.00.

Table 26: Results for the Altered Constrained Eigenstructure Assignment Method Model Updating Method on Full Scale Model

Modes	Analytical Eigenvalues	Measured Eigenvalues	Updated System Eigenvalues	Percent Error (%)	MAC
1	17.01	17.55	17.01	3.07	1.00
2	33.33	34.08	33.33	2.22	1.00
3	77.79	78.13	77.66	0.63	1.00
4	294.93	295.49	295.07	0.14	1.00
5	358.36	359.33	358.34	0.27	1.00
6	446.49	447.02	446.38	0.14	1.00
7	832.67	833.97	833.08	0.11	1.00
8	1460.90	1462.18	1460.92	0.09	1.00
9	1779.90	1778.80	1777.30	0.08	1.00
10	3205.60	3205.64	3204.76	0.03	1.00
11	4018.80	4021.76	4019.52	0.06	1.00
12	4951.80	4952.40	4952.13	0.02	1.00
13	5705.00	5704.82	5705.83	0.02	1.00
14	5705.00	5713.55	5708.24	0.09	1.00
15	7551.10	7559.01	7555.40	0.05	1.00
16	8200.40	8197.37	8195.69	0.02	1.00
17	9759.80	9756.42	9754.49	0.02	1.00
18	11179.00	11194.39	11192.61	0.01	1.00

For the reduced order model, the Altered Constrained Eigenstructure Assignment method had the same issues for parameter selection, since both the mass and stiffness matrices are filled. Table 27 displays the results for the Altered Constrained Eigenstructure Assignment method. However, this method produced better results, as only one MAC value is below 0.95 and the eigenvalue errors are relatively small.

Table 27: Results for Altered Constrained Eigenstructure Assignment Method Model Updating on Order-Reduced Full Scale Model

Modes	Analytical Eigenvalues	Measured Eigenvalues	Updated System Eigenvalues	Percent Error (%)	MAC
1	17.01	17.55	17.94	2.21	0.99
2	33.33	34.08	31.82	6.63	1.00
3	77.79	78.13	78.29	0.17	1.00
4	294.93	295.49	294.34	0.40	1.00
5	358.36	359.33	358.30	0.29	1.00
6	446.49	447.02	435.59	2.55	1.00
7	832.67	833.97	832.92	0.13	0.99
8	1460.90	1462.18	1481.92	1.34	0.93
9	1779.90	1778.80	1762.18	0.93	0.99
10	3205.60	3205.64	3216.98	0.35	1.00
11	4018.80	4021.76	4019.42	0.06	1.00
12	4951.80	4952.40	4974.55	0.43	1.00
13	5705.00	5704.82	5681.40	0.45	0.95
14	5705.00	5713.55	5741.10	0.54	0.95
15	7551.10	7559.01	7556.29	0.03	1.00
16	8200.40	8197.37	8189.75	0.09	0.99
17	9759.80	9756.42	9748.50	0.08	1.00
18	11179.00	11194.39	11203.39	0.09	1.00

4.9 Summary

The full scale model example provides a better understanding of the advantages and disadvantages inherit in each model updating method.

The Lagrange multiplier methods and the state-space ESA method have the ability to update the structural system quickly and accurately, even in complex situations. The penalty function method is also able to reproduce accurate solutions, however, it is more computationally demanding.

However, the quadratic pencil method, the constrained ESA method, and the altered constrained ESA method all have difficulties with the order-reduced full scale model. However, even with those difficulties, the altered ESA method is able to reproduce the desired eigenvalues and eigenvectors with good accuracy.

Table 28 is a numerical summary of each methods performance when applied to the full scale model. It displays the average eigenvalue error, the largest eigenvalue error, the average MAC value, and the lowest MAC value. Recall that the state space method was reduced to six modes. The constrained ESA method is the only one to have a MAC value lower than 0.95.

Table 28: Summary of Method Performances on Full Scale Model

Method	Average Eigenvalue Error (%)	Largest Eigenvalue Error (%)	Average MAC	Lowest MAC
Lagrange Multiplier (Stiffness Matrix Updated)	0.00	0.00	1.00	1.00
Lagrange Multiplier (Stiffness and Mass Matrices Updated)	0.00	0.00	1.00	1.00
Penalty Function Method	0.07	0.54	1.00	1.00
Quadratic Pencil Method	0.00	0.03	0.99	0.95
State Space Method	0.00	0.00	0.99	0.95
Constrained ESA Method	0.24	2.61	0.99	0.77
Altered Constrained ESA Method	0.39	3.07	1.00	1.00

Table 29 is a numeral summary of the methods performances on the order-reduced full scale model. The same values are compared as Table 28.

Table 29: Summary of Method Performances on Order-Reduced Full Scale Model

Method	Average Eigenvalue Error (%)	Largest Eigenvalue Error (%)	Average MAC	Lowest MAC
Lagrange Multiplier (Stiffness Matrix Updated)	0.00	0.00	1.00	1.00
Lagrange Multiplier (Stiffness and Mass Matrices Updated)	0.00	0.00	1.00	1.00
Penalty Function Method	1.90	11.22	0.99	0.91
Quadratic Pencil Method	6.21	24.31	0.63	0.00
State Space Method	0.00	0.00	0.99	0.95
Constrained ESA Method	2.00	9.03	0.87	0.42
Altered Constrained ESA Method	0.93	6.63	0.99	0.93

Note that, since the state space method could only be done by reducing the full scale model to six degrees of freedom the results the same as the previous table. The quadratic pencil method and the constrained ESA method have large average eigenvalue error and average MAC values below 0.95.

5.0 Conclusion

The advantages and disadvantages of each method vary significantly and suggest that different finite-element model situations will require different methods. In order to highlight some important conclusions, three separate situations will be introduced: Simple and Small, Simple and Large, and Complex.

5.1 Simple and Small

The “small” situation will be defined as a model with ten or less degrees of freedom and the “simple” situation will be defined as a model with a banded stiffness matrix and a diagonal mass matrix. For this scenario, it is suggested that the Altered Eigenstructure Assignment method is the most effective. Since both of the stiffness and mass matrices will have physical meaning, it is important to preserve their forms. This method is also the only method that retains the physical representation of the matrices and updates the mass matrix in addition to the damping matrix. The mass matrix is believed to be a more reasonable matrix to update over the damping matrix, since damping is unmeasured and often unknown. It should be noted that when modal parameters are being extracted through vibration testing, they are real values, and therefore, are not damped responses. Finally, this method is capable of reproducing near exact modal parameters for small and simple FE models.

5.2 Simple and Large

The “simple and large” situation is defined as a FEM that contains more than ten degrees of freedom and has a banded stiffness matrix and a diagonal mass matrix. For this situation, it is suggest that the Quadratic Pencil method is the most effective. Throughout the examples this method was applied to, it was shown to reproduce the measured modal parameters example. For large models, the Altered Constrained Eigenstructure Assignment method was not as reliable, required an increase in calculations, and displayed a trend of losing accuracy proportionally to the increase of update parameters; whereas the Quadratic Pencil method requires the same calculations, which are simple and fast.

However, if physical representation of the updated FEM is very important, the Quadratic Pencil method does not guarantee the conservation of the analytical form. In that situation, the Penalty Function method is suggested. Although it is not as accurate as the Quadratic Pencil method, it does reproduce the modal parameters almost perfectly and conserves the form of the analytical matrices, since the parameters are chosen by the user.

5.3 Complex

The “complex” situation is defined as a FE model that has, at least, a filled stiffness matrix. If the FE model also has a filled mass matrix the Raw Measured Data Direct Updating method is suggested. This method is capable of closely reproducing all of the modal parameters, and there is no requirement to either define update parameters and sensitivities, or to conserve the physical representation of the mass and stiffness matrices. The Indirect methods would require a definition of update parameters, which can be difficult to obtain when dealing with filled matrices, especially for large number of degrees of freedom.

However, if the mass matrix is diagonal, which was the case for the full scale model, the Quadratic Pencil method is suggested. This method does not update the mass matrix, only the damping and stiffness matrices, leaving the mass matrix in its diagonal form. Also, since the stiffness matrix is filled, the fact that this method can fill the stiffness method is of little consequence.

Finally, the Altered Constrained Eigenvalue Assignment method can still be used, but only if there are specific modes that are required to be updated. However, it is only reasonable if the engineer has good knowledge of the matrices and if the mass and stiffness values that affect the modes that are to be updated are known.

6.0 Recommended Future Work

One of the major problems that presented itself during this thesis was that some of the methods were affected by the form of the identified eigenvectors. I think that it would be beneficial to be able to have a method in place that could convert the form of the eigenvector.

Since updating parameter sensitivities are not objectively found, a study that could more clearly define the sensitivities of each updating parameter would be very helpful for the penalty function method, the constrained ESA method, and the altered ESA method.

I believe that further research on control-based methods in order to improve their accuracy and reduce their computational completely is important, since these methods are very powerful.

References

Andry, A. N., Shapiro, E. Y., & Chung, J. C. (1983). Eigenstructure Assignment For Linear Systems. *IEEE Transactions on Aerospace and Electronic Systems*, 5, 711-729.

Bagchi, A. (2005). Updating the Mathematical Model of a Structure Using Vibration Data. *Journal of Vibration and Control*, 11(12), 1469-1486.

Baruch,M. (1978). Optimization Procedure to Correct Stiffness and Flexibility Matrices Using Vibration Data. *AIAA Journal*, 16, 1208-1210.

Berman, A. & Nagy, E. J. (1983). Improvement of a Large Analytical Model Using Test Data. *AIAA Journal*, 21, 1168-1173.

Brownjohn, J. M. W., Xia, P., Hao, H., & Xia, Y. (2001). Civil Structure Condition Assessment By FE Model Updating: Methodology And Case Studies. *Finite-elements in Analysis and Design*, 37, 761-775.

Cunha, J., Foltete, E., & Bouhaddi, N. (2008). Evaluation Of Stiffness Of Semi-rigid Joints In Pultruded Profiles From Dynamic And Static Data By Using Model Updating Technique. *Engineering Structure*, 30, 1024-1036

Daniell, W. E., & Macdonald, J. H. G. (2007). Improved Finite-element Modelling Of A Cable-stayed Bridge Through Systematic Manual Tuning. *Engineering Structures*, 29, 358-371.

Datta, B. N., Elhay, S., Ram, Y. M., & Sarkissian, D. R. (2000). Partial Eigenstructure Assignment For The Quadratic Pencil. *Journal of Sound and Vibration*, 203, 101-110.

Datta, B. N. (2002). Finite-Element Model Updating, Eigenstructure Assignment And Eigenvalue Embedding Techniques For Vibrating Systems. *Mechanical Systems and Signal Processing*, 16, 83-96.

Friswell, M. I., & Mottershead,J. E.(1995). *Finite-element Model Updating in Structural Dynamics.* Dordrecht, The Netherlands: Kluwear Academic Publishers.

Friswell, M. I., & Garvey, S. D., & Penny, P. E. (1995). Model Reduction Using Dynamic and Iterated IRS Techniques. *Journal of Sound and Vibration*, 186(2), 311-323.

Friswell, M. I., & Garvey, S. D., & Penny, P. E. (1998). The Convergence of the Iterated IRS Method. *Journal of Sound and Vibration*,211, 123-132.

Friswell, M. I., & Garvey, S. D., & Penny, P. E. (1996). The Application of the IRS and Balanced Realization Methods to Obtain Reduced Models of Structures with Local Non-Linearities. *Journal of Sound and Vibration*, 196(4), 453-468.

Fiswell, M.I., Mottershead, J.E (1993). Model Updating In Structural Dynamics: A Survey. *Journal of Sound and Vibration*, 167(2), 347-375.

Fox, R. L. & Kapoor, M. P. (1968). Rate Of Change Of Eigenvalues And Eigenvectors. *AIAA Journal*, 12(6), 2426-2429.

Inman, D. J. & Schulz, M. J. (1994). Model Updating Using Constrained Eigenstructure Assignment. *Journal of Sound and Vibration*, 178, 113-130.

Miguel, L. F. F., Miguel, L. F. F., & Menezes, R. C. R. (2006). Model Updating Of A Frame Structure Using Penalty Functions Based Procedures. *Mecanica Computacional*, 25, 1583-1592.

Moore, B. C. (1976). On the Flexibility Offered by State Feedback in Multivariable Systems Beyond Closed Loop Eigenvalue Assignment. *IEEE Trans. Automat. Contr.*, 21, 689-692.

Nelson, R. B. (1976). Simplified Calculation of Eigenvector Derivatives. *AIAA Journal*, 14, 1201-1205.

O'Callahan, J, Avitabile, P, and Riemer, R (1989). System Equivalent Reduction Expansion Process (SEREP). Seventh International Modal Analysis Conference, Las Vegas, Nevada.

Schlune, H, Plos, M., & Gylltoft,K. (2009). Improved Bridge Evaluation Through Finite-element Model Updating Using Static and Dynamic Measurements. *Engineering Structures*, 1477-1485.

Titurus, B, Friswell, M. I., & Starek, L. (2003). Damage Detection Using Generic Elements: Part I. Model Updating. *Computers and Structures*, 81, 2273-2286.

Wittrick, W. H. (1962). Rates of Change of Eigenvalues, with Reference to Buckling and Vibration Problems. *Journal of the Royal Aeronautical Society*, 66, 590-591.

APPENDIX

A1 – Guyan or Static Model Reduction

```
% Guyan or Static Reduction

% Written by Paul E Paquet
% July 6, 2009

%Structural Dynamics, Control, and Identification
%Department of Civil and Environmental Engineering
%University of Waterloo, Waterloo, Ontario, Canada

% Remove all other variable and clear the command window.
clear all
clc

% Running AMod to get the Mass and Stiffness matrices of the structural
% model.
AMod

% Re-arranging the stiffness and mass matrices into four sections
% the mm (master-master), ss (slave-slave), sm (slave-master) and ms
% (master-slave) in preparation to reduce the sm and ss sections.
Kmm = [K(1:2,1:2) K(1:2,5:6) ; K(5:6,1:2) K(5:6,5:6)];
Kss = [K(3:4,3:4) K(3:4,7:8) ; K(7:8,3:4) K(7:8,7:8)];
Ksm = [K(3:4,1:2) K(3:4,5:6) ; K(7:8,1:2) K(7:8,5:6)];
Kms = Ksm';

Mmm = [M(1:2,1:2) M(1:2,5:6) ; M(5:6,1:2) M(5:6,5:6)];
Mss = [M(3:4,3:4) M(3:4,7:8) ; M(7:8,3:4) M(7:8,7:8)];
Msm = [M(3:4,1:2) M(3:4,5:6) ; M(7:8,1:2) M(7:8,5:6)];
Mms = Msm';

% Creating the transfer matrix that will reduce the matrices
Ts = [eye(4) ; -inv(Kss) * Ksm];

% Placing the matric sections in the appropriate spots.
Ma = [Mmm Mms ; Msm Mss];
Ka = [Kmm Kms ; Ksm Kss];

% Calculating the reduced mass and stiffness matrices.
Mr = Ts' * Ma * Ts;
Kr = Ts' * Ka * Ts;

% Calculating, both, the reduced and initial eigenvalues and eigenvectors
% in order to compare them to ensure the correct values were preserved.
[V D] = eig(Ka,Ma);
[Vr Dr] = eig(Kr,Mr);

% Clear out unwanted variables.
clear Mmm Mms Msm Mss Kmm Kms Ksm Kss M K
```

A2 – Improved Reduction System

```
% Improved Reduction System Model Reduction

% Written by Paul E Paquet
% July 6, 2009

%Structural Dynamics, Control, and Identification
%Department of Civil and Environmental Engineering
%University of Waterloo, Waterloo, Ontario, Canada

% Remove all other variable and clear the command window.
clear all
clc

% Running AMod to get the Mass and Stiffness matrices of the structural
% model.
AMod

Mo = 8;

% Complete four IRS reductions.
for j = 1 : 4
    % Calculate k/m for the diagonal terms
    for i = 1 : Mo
        KM(i) = K(i,i) / M(i,i);
    end

    % Find the maximum value and its index
    [MM In] = max(KM);

    % Re-organize the mass and stiffness matrices accordingly
    for i = 1 : Mo
        if i < In
            for k = 1 : Mo
                if k < In
                    Kmm(i,k) = K(i,k);
                    Mmm(i,k) = M(i,k);
                elseif k > In
                    Kmm(i,k-1) = K(i,k);
                    Mmm(i,k-1) = M(i,k);
                else
                    Kms(i,1) = K(i,k);
                    Mms(i,1) = M(i,k);
                end
            end
        elseif i > In
            for k = 1 : Mo
                if k < In
                    Kmm(i-1,k) = K(i,k);
                    Mmm(i-1,k) = M(i,k);
                elseif k > In
                    Kmm(i-1,k-1) = K(i,k);
                    Mmm(i-1,k-1) = M(i,k);
                else
                    Kms(i-1,1) = K(i,k);
                    Mms(i-1,1) = M(i,k);
```

```
                end
            end
        else
            for k = 1 : Mo
                if k < In
                    Ksm(1,k) = K(In,k);
                    Msm(1,k) = M(In,k);
                elseif k > In
                    Ksm(1,k-1) = K(In,k);
                    Msm(1,k-1) = M(In,k);
                else
                    Kss = K(In,k);
                    Mss = M(In,k);
                end
            end
        end
    end

    K = [Kmm Kms;Ksm Kss];
    M = [Mmm Mms;Msm Mss];

    % Set up the Static transfomation matrix
    S = [zeros(Mo-1, Mo) ; zeros(1,Mo-1) inv(Kss)];
    Ts = [eye(Mo-1); -inv(Kss)*Ksm];

    % Set up statically reduced mass and stiffness matrices
    Kr = Ts' * K * Ts;
    Mr = Ts' * M * Ts;

    % Calculate the IRS transformation matrix
    Ti = Ts + S * M * Ts * inv(Mr) * Kr;

    % Calucluate the IRS reduced mass and stiffness matrices
    Mi = Ti' * M * Ti;
    Ki = Ti' * K * Ti;

    % Set M and K to the newly reduced matrices in order to use the
    % variables for tne next reduction
    M = Mi;
    K = Ki;

    % Clear variables to ensure that no errors are caused by them
    clear In KM Ki Kmm Kms Kr Ksm Kss MM Mi Mmm Mr Msm Mss Mms S Ti Ts
    % Set matrix dimensions to one less
    Mo = Mo - 1;
end
```

A3 – Iterated Improved reduction System

```
% Improved Reduction System Model Reduction

% Written by Paul E Paquet
% July 6, 2009

%Structural Dynamics, Control, and Identification
%Department of Civil and Environmental Engineering
%University of Waterloo, Waterloo, Ontario, Canada

% Remove all other variable and clear the command window.
clear all
clc

% Running AMod to get the Mass and Stiffness matrices of the structural
% model.
AMod

Mo = 8;

% Complete four IRS reductions.
for j = 1 : 4
    % Calculate k/m for the diagonal terms
    for i = 1 : Mo
        KM(i) = K(i,i) / M(i,i);
    end

    % Find the maximum value and its index
    [MM In] = max(KM);

    % Re-organize the mass and stiffness matrices accordingly
    for i = 1 : Mo
        if i < In
            for k = 1 : Mo
                if k < In
                    Kmm(i,k) = K(i,k);
                    Mmm(i,k) = M(i,k);
                elseif k > In
                    Kmm(i,k-1) = K(i,k);
                    Mmm(i,k-1) = M(i,k);
                else
                    Kms(i,1) = K(i,k);
                    Mms(i,1) = M(i,k);
                end
            end
        elseif i > In
            for k = 1 : Mo
                if k < In
                    Kmm(i-1,k) = K(i,k);
                    Mmm(i-1,k) = M(i,k);
                elseif k > In
                    Kmm(i-1,k-1) = K(i,k);
                    Mmm(i-1,k-1) = M(i,k);
                else
                    Kms(i-1,1) = K(i,k);
                    Mms(i-1,1) = M(i,k);
```

```
                end
            end
        else
            for k = 1 : Mo
                if k < In
                    Ksm(1,k) = K(In,k);
                    Msm(1,k) = M(In,k);
                elseif k > In
                    Ksm(1,k-1) = K(In,k);
                    Msm(1,k-1) = M(In,k);
                else
                    Kss = K(In,k);
                    Mss = M(In,k);
                end
            end
        end
    end

    K = [Kmm Kms;Ksm Kss];
    M = [Mmm Mms;Msm Mss];

    % Set up the Static transfomation matrix
    S = [zeros(Mo-1, Mo) ; zeros(1,Mo-1) inv(Kss)];
    Ts = [eye(Mo-1); -inv(Kss)*Ksm];

    % Set up statically reduced mass and stiffness matrices
    Kr = Ts' * K * Ts;
    Mr = Ts' * M * Ts;

    % Calculate the IRS transformation matrix
    Ti = Ts + S * M * Ts * inv(Mr) * Kr;

    % Calucluate the IRS reduced mass and stiffness matrices
    Mi = Ti' * M * Ti;
    Ki = Ti' * K * Ti;

    % Complete the Iterated IRS transformation
    ts = -inv(Kss) * Ksm;
    for i = 1:100
        ti = ts + inv(Kss) * [Msm Mss] * Ti * pinv(Mr) * Kr;
        Ti = [eye(4) ; ti];
        Mr = Ti' * M * Ti;
        Kr = Ti' * K * Ti;
    end

    % Set M and K to the newly reduced matrices in order to use the
    % variables for tne next reduction
    M = Mi;
    K = Ki;

    % Clear variables to ensure that no errors are caused by them
    clear In KM Ki Kmm Kms Kr Ksm Kss MM Mi Mmm Mr Msm Mss Mms S Ti Ts
    % Set matrix dimensions to one less
    Mo = Mo - 1;
end
```

A4 – System Equivalent reduction Expansion Process (SEREP)

```
function [KR,MR,EigVec,EigVal,EigVecRed,EigValRed] = SEREP(K,M,mDOF,MoI)
%SEREP System Equivalent Reduction Expansion Process
%   Produces a reduced model which preserves the dynamic character of the
%   original full system model for the selected modes of interest.
%   Inputs: full system model stiffness matrix (K), full system model mass
%    matrix (M), chosen master degress of freedom listed in a single row
(mDOF),
%   chosen modes of interest listed in a single row (MoI)
%   Outputs: reduced model stiffness matrix (KR), reduced model mass
%   matrix (MR), full model eigenvector matrix (EigVec), full model
%   eigenvalue matrix (EigVal), reduced model eigenvector matrix
%   (EigVecRed), reduced model eigenvalue matrix (EigValRed)
%
%   Developed by Aaron J Roffel
%   August 25, 2008
%
%   Structural Dynamics, Control and Identification
%   Department of Civil and Environmental Engineering
%   University of Waterloo, Waterloo, Ontario, Canada
%
%   References:
%   O'Callahan, J., Avitabile, P., & Riemer, R. System equivalent reduction
%   expansion process (SEREP). Las Vegas, NV (USA).

[EigVec,EigVal]=eig(K,M);

nDOF=size(K,2);
nM=size(EigVal,2);

L=zeros(1,nM);

for k=1:nM
    L(1,k)=EigVal(k,k);
end

[L,index]=sort(L);

EigVecTemp=EigVec;
EigValTemp=EigVal;

for k=1:nM
    EigVecTemp(:,k)=EigVec(:,index(k));
    EigValTemp(k,k)=L(1,k);
end

EigVec=EigVecTemp;
EigVal=EigValTemp;

clear EigVecTemp EigValTemp index L k

nmDOF=size(mDOF,2);
nMoI=size(MoI,2);

sDOF=1:length(K(:,1));
```

```
sDOF(mDOF) = [];

Um=EigVec(mDOF,MoI);
Us=EigVec(sDOF,MoI);

if nMoI>nmDOF
    error('Number of modes of interest exceeds number of chosen master degrees of freedom. Choose master degrees of freedom such that the number chosen is greater than or equal to the number of modes of interest')
end

if nMoI<nmDOF
    T=[Um; Us]*(Um'*Um)^-1*Um';
else
    T=[Um; Us]*Um^-1;
end

MR=T'*M*T;
KR=T'*K*T;

[EigVecRed,EigValRed]=eig(KR,MR);

L=zeros(1,nMoI);

for k=1:nMoI
    L(1,k)=EigValRed(k,k);
end

[L,index]=sort(L);

EigVecRedTemp=EigVecRed;
EigValRedTemp=EigValRed;

for k=1:nMoI
    EigVecRedTemp(:,k)=EigVecRed(:,index(k));
    EigValRedTemp(k,k)=L(1,k);
end

EigVecRed=EigVecRedTemp;
EigValRed=EigValRedTemp;
```

clear EigVecRedTemp EigValRedTemp index L k

B1 – Correction of Stiffness Matrix (Direct Model Updating)

```
% Direct Updating Using Updated Eigenvectors

% Written by Paul E Paquet
% July 6, 2009

%Structural Dynamics, Control, and Identification
%Department of Civil and Environmental Engineering
%University of Waterloo, Waterloo, Ontario, Canada

% Remove all other variable and clear the command window.
clear all
clc

% Run script that gives the analytical mass and stiffness matrices
% as well as the measured eigenvalues and eigenvectors.
ModProp

% Calculate the updated eigenvector matrix.
Vu = V_mea / sqrtm(V_mea' * M * V_mea);

% Using Vu, calculate each term in the stiffness updating equation.
K1 = K * Vu * Vu' * M;
K2 = M * Vu * Vu' * K;
K3 = M * Vu * Vu' * K * Vu * Vu' * M;
K4 = M * Vu * Lm * Vu' * M;

% Put the together.
Ku = K - K1 - K2 + K3 + K4;

% Remove all near zero terms inside the mass and stiffness matrices in
% order to avoid small perturbations in eigenvalues or eigenvectors
for i = 1 : 4
    for k = 1 : 4
        if abs(Ku(i,k)) < 0.0001
            Ku(i,k) = 0;
        end
    end
end

% Find the updated eigensystem.
[V_new L_new] = eig(Ku,M);

% Put the eigenvalues into a vector and sort them.
l = diag(L_new);
[l Id] = sort(l);

% Calculate the MAC value of the eigenvectors.
for i = 1 : 4
    Mac(i) = ((V_new(:,i)' * V_mea(:,Id(i)))^2) / ((V_new(:,i)' * V_new(:,i))
* (V_mea(:,Id(i))' * V_mea(:,Id(i))));
end
```

B2 – Direct Updating Using Raw Measured Data

```
% Raw Measured Data Direct Method Model Updating

% Written by Paul E Paquet
% July 6, 2009

%Structural Dynamics, Control, and Identification
%Department of Civil and Environmental Engineering
%University of Waterloo, Waterloo, Ontario, Canada

% Remove all other variable and clear the command window.
clear all
clc

% Run script that gives the analytical mass and stiffness matrices
% as well as the measured eigenvalues and eigenvectors.
Modprop

% Calculate the values of some variables
Mahat = Vm' * M * Vm;
R = eye(4) - Mahat;
IM = inv(Mahat);

% Calculate the updated mass matrix
Mu = M + M * Vm * IM * R * IM * Vm' * M;

% Break the stiffness updating equation into four terms.
K1 = K * Vm * Vm' * Mu;
K2 = Mu * Vm * Vm' * r;
K3 = Mu * Vm * Vm' * K * Vm * Vm' * Mu;
K4 = Mu * Vm * Lm * Vm' * Mu;

% Sum them together to find the updated stiffness matrix.
Ku = K - K1 - K2 + K3 + K4;

% Remove variables that are no longer necessary.
clear K1 K2 K3 K4 Mahat IM R

% Remove all near zero terms inside the mass and stiffness matrices in
% order to avoid small perturbations in eigenvalues or eigenvectors
for i = 1 : 4
    for k = 1 : 4
        if abs(Ku(i,k)) < 0.001
            Ku(i,k) = 0;
        end
        if Mu(i,k) < 0.00001
            Mu(i,k) = 0;
        end
    end
end

% Find the updated eigenvalues and eigenvectors of the system.
[Vu Du] = eig(Ku,Mu);

% Find the MAC value of each mode shape.
```

```
for i = 1 : 4
    Mac(i) = ((Vu(:,i)' * Vm(:,i))^2) / ((Vu(:,i)' * Vu(:,i)) * (Vm(:,i)' *
Vm(:,i)));
end
```

B3 – Penalty Function Model Updating

```
% Penalty Function Model Updating Method

% Written by Paul E Paquet
% July 6, 2009

%Structural Dynamics, Control, and Identification
%Department of Civil and Environmental Engineering
%University of Waterloo, Waterloo, Ontario, Canada

% Remove all other variable and clear the command window.
clear all
clc

% Retrieve the analytical mass and stiffness matrices; as well as the
% measured eigenvalues and eigenvectors.
ModProp

% Create matrices for the parameters
for i = 1 : 7
    dK(:,:,i) = zeros(4);
end

% Assign parameter derivatives.
dK(1,1,1) = 1;
dK(1,2,2) = 1;
dK(2,1,2) = 1;
dK(2,2,3) = 1;
dK(2,3,4) = 1;
dK(3,2,4) = 1;
dK(3,3,5) = 1;
dK(3,4,6) = 1;
dK(4,3,6) = 1;
dK(4,4,7) = 1;

% Set up the initial parameter changes
P = zeros(7,1);

% Use 5 iterations of parameter updating
for j = 1 : 5
    for i = 1 : 7 % each row is a property (eigenvalue-vector)
        for k = 1 : 4 % each column is a parameter
            dl(k,i) = V(:,k)' * (dK(:,:,i) - d(k) * dM(:,:,i)) * V(:,k);
        end
    end

    % Sensitivity matrix
    S = dl;

    % Calculate the change in parameters
    P_1 = P + S' * pinv(S * S') * (d_mea - d);

    % Make an updating stiffness matrix
    Ku1 = zeros(4);
    for b = 1 : 7
```

```
        Ku1 = Ku1 + P_1(b) * dK(:,:,b);
    end

    % Update the stiffness matrix
    Ku = K + Ku1;
    % Calculate the new eigenvectors and eigenvalues
    [Vu Du] = eig(Ku,M);

    K = Ku;
    d = diag(Du);
    V = Vu;
end

% Calculate the MAC values
for i = 1 : 4
    Mac(i)  =  ((Vu(:,i)'  *  V_mea(:,i))^2)  /  ((Vu(:,i)'  *  Vu(:,i))  *
(V_mea(:,i)' * V_mea(:,i)));
end
```

B4 – State Space Model Updating Method

```matlab
% State Space Full State Feedback on the 4 Storey Model

% Written by Paul E Paquet
% July 6, 2009

%Structural Dynamics, Control, and Identification
%Department of Civil and Environmental Engineering
%University of Waterloo, Waterloo, Ontario, Canada

% Remove all other variable and clear the command window.
clear all
clc

% Call the script that contains the analytical dynamic properties and the
% measured modal properties.
ModProp

% Setting up the state space matrix.
A = [zeros(4) eye(4) ; -inv(M) * Kr, -inv(M) * D];

% Do the updating using 13 different B matrices
for k = 1 : 13
    B = zeros(8);
    if k == 1
        B(5:8,1:4) = eye(4);
        B(5:8,5:8) = eye(4);
    elseif k == 2
        B = eye(8);
    elseif k == 3
        B(5:8,5:8) = eye(4);
    elseif k == 4
        B(5:8,1:4) = eye(4);
    elseif k == 5
        B(5:6,1:2) = eye(2);
    elseif k == 6
        B(5:6,5:6) = eye(2);
    elseif k == 7
        B(5,5) = 1;
        B(5,1) = 1;
    elseif k == 8
        B(5:6,1:2) = eye(2);
        B(5:6,5:6) = eye(2);
    elseif k == 9
        B(7:8,3:4) = eye(2);
        B(7:8,7:8) = eye(2);
    elseif k == 10
        B(1:4,1:4) = eye(4);
        B(1:4,5:8) = eye(4);
    elseif  k == 11
        B(1:4,1:4) = eye(4);
        B(1:4,5:8) = eye(4);
        B(5:8,1:4) = eye(4);
        B(5:8,5:8) = eye(4);
    elseif k == 12
        B = ones(8);
```

```
    else
        B(1:4,5:8) = eye(4);
        B(5:8,1:4) = eye(4);
        B(5:8,5:8) = eye(4);
    end

    % Check the rank of B to see if there is correlation
    P(k) = rank(B);

    % Complete the calculations for MZ and the desired eigenvectors
    for i = 1 : 8
        S(:,:,i) = [(L_mea(i,i) * eye(8)) - A , B];
        R(:,:,i) = null(S(:,:,i));
        N_l(:,:,i) = R(1:8,:,i);
        M_l(:,:,i) = R(9:16,:,i);
        V_des(:,i) = 45 * N_l(:,1,i) + 4 * N_l(:,2,i) + 4 * N_l(:,3,i) + 1.5
* N_l(:,4,i) + 1.25 * N_l(:,5,i) + 1.5 * N_l(:,6,i) + 2 * N_l(:,7,i) + 1.75 *
N_l(:,8,i);
        z(:,:,i) = pinv(N_l(:,:,i)) * V_des(:,i);
        MZ(:,:,i) = -M_l(:,:,i) * z(:,:,i);
    end

    % Remove complex values
    for j = 1 : 8
        if mod(j,2) == 1
            V_d_(:,j) = real(V_des(:,j));
            MZ_(:,j) = real(MZ(:,1,j));
        else
            V_d_(:,j) = imag(V_des(:,j-1));
            MZ_(:,j) = imag(MZ(:,1,j-1));
        end
    end

    % Calculate the gain matrix
    K = MZ_ * pinv(V_d_);

    % Update the State matrix and find the updated eigenvector and
    % eigenvalue matrices
    A_new(:,:,k) = A + B * K;
    [V_new(:,:,k) D_new(:,:,k)] = eig(A_new(:,:,k));

    % Sort the eigenvalues so that they are in the same order
    d_new = diag(D_new(:,:,k));
    [d_new Idn] = sort(d_new);
    l_mea = diag(L_mea);
    [l_mea Idm] = sort(l_mea);

    % Put the eigenvalues into real terms
    d_new = [d_new(1) * d_new(2); d_new(3) * d_new(4); d_new(5) * d_new(6);
d_new(7) * d_new(8)];
    d_new_(:,k) = d_new;
    l_mea = [l_mea(1) * l_mea(2); l_mea(3) * l_mea(4); l_mea(5) * l_mea(6);
l_mea(7) * l_mea(8)];

    % Organize the eigenvectors into real form that is comparable to the
```

```
    % other methods
    a = 1;
    for i = 1 : 4
        V_m(:,i) = V_mea(:,a) .* V_mea(:,a+1);
        V_n(:,i) = V_new(:,a) .* V_new(:,a+1);
        a = a + 2;
    end
    V_m = V_m(1:4,:) + V_m(5:8,:);
    V_n = V_n(1:4,:) + V_n(5:8,:);

    % Calculate the MAC values
    for i = 1 : 4
        MAC(i) = ((V_m(:,i)' * V_n(:,i))^2) / ((V_m(:,i)' * V_m(:,i)) *
(V_n(:,i)' * V_n(:,i)));
    end
end
```

B5 – Quadratic Pencil Method

```
% Quadratic Pencil Updating Method

% Written by Paul E Paquet
% July 6, 2009

%Structural Dynamics, Control, and Identification
%Department of Civil and Environmental Engineering
%University of Waterloo, Waterloo, Ontario, Canada

% Remove all other variable and clear the command window.
clear all
clc

% Retrieve the analytical mass and stiffness matrices; as well as the
% measured eigenvalues and eigenvectors.
ModProp

% Calculate the variables defined by Datta
Z = L_mea * V_mea' * M * V * L - V_mea' * K * V;
B_hat = M * V_mea * (L_mea .^ 2) + C * V_mea * L_mea + K * V_mea;
F_hat = M * V * L * inv(Z);
G_hat = - K * V * inv(Z);

X = B_hat * [F_hat' G_hat'];

% Complete the SVD of the matrix X. Econ is used so that the dimensions of
% the sigular value decomposition are correct
[U,S,V1] = svd(X,'econ');

% Get the values of the components that make up the two gain matrices.
B = real(U * S);
F = real(V1(1:4,:));
G = real(V1(5:8,:));

% Remove any near zero values to avoid incorrect updating
for i = 1 : 4
    for k = 1 : 4
        if abs(F(i,k)) < 0.001
            F(i,k) = 0;
        end
        if abs(G(i,k)) < 0.001
            G(i,k) = 0;
        end
    end
end

% Update the damping and stiffness matrices using the gain matrices
Cu = C - B * F';
Ku = K - B * G';

% Remove any near zero values to avoid major changes to the eigenvalues and
% eigenvectors
for i = 1 : 4
    for k = 1 : 4
```

```
        if abs(Ku(i,k)) < 0.001
            Ku(i,k) = 0;
        end
        if abs(Cu(i,k)) < 0.001
            Cu(i,k) = 0;
        end
    end
end

% Find the updated eigenvectors and eigenvalues
[Vu Du] = polyeig(Ku,Cu,M);

% Re-order the eigenvalues and eigenvectors so that they match the order of
% the measured values.
[du Im] = sort(Du);
for i = 1 : 8
    TempVm(:,i) = V_mea(:,Im(i));
    TempVu(:,i) = Vu(:,Id(i));
end
V_mea = TempVm;
Vu = TempVu;

a = 1;
for i = 1 : 4
    Vu_r(:,i) = Vu(:,a) .* Vu(:,a+1);
    V_mea_r(:,i) = V_mea(:,a) .* V_mea(:,a+1);
    a = a + 2;
end

% Find MAC values
a = 1;
for i = 1 : 4
    Mac(i) = ((Vu_r(:,i)' * V_mea_r(:,i)) ^ 2) / ((Vu_r(:,i)' * Vu_r(:,i)) *
(V_mea_r(:,i)' * V_mea_r(:,i)));
    du_r(i) = du(a) * du(a+1);
    d_mea_r(i) = d_mea(a) * d_mea(a+1);
    a = a + 2;
end
```

B6 – Constrained Eigenstructure Assignment

```
% Constrained Eigenstructure Assignment function script

% Written by Paul E Paquet
% July 6, 2009

%Structural Dynamics, Control, and Identification
%Department of Civil and Environmental Engineering
%University of Waterloo, Waterloo, Ontario, Canada

% Define the function that produces the optimization cost function f.
function [f,dJ] = InTwoStorey_C(x,G,H)

% Calling the properties for the 4 storey model (included are the new
% eigenvalues and eigenvectors
ModProp

% Find the needed values for 2 measured eigenvalues.
for i = 1 : 2
    Gamma(:,:,i) = [(M * (L_mea(i) ^ 2) + C * L_mea(i) + K) eye(4)];
    Vi(:,:,i) = null(Gamma(:,:,i),'r');
    Vik(:,:,i) = Vi(1:4,:,i);
    V_hat(:,:,i) = Vi(5:8,:,i);
    e(:,:,i) = (inv(Vik(:,:,i))) * V_mea(1:4,i);
end

% Ensuring that the matrices have the proper dimensions
E = zeros(8,2);
V(:,:) = zeros(4,8);
V_H(:,:) = zeros(4,8);

V(:,:) = [Vik(:,:,1) Vik(:,:,2)];
V_H(:,:) = [V_hat(:,:,1) V_hat(:,:,2)];

for i = 1 : 2
    E(1:4,i) = e(:,:,i);
end

% Setting up the matrices G and H by defining and calculating all of the
% values used.
Le = diag(L_mea);
Le_r = real(Le);
Le_i = imag(Le);

VE_r(:,:) = real(V*E);
VE_i(:,:) = imag(V*E);

G11 = VE_r * Le_r - VE_i * Le_i;
G12 = VE_r * Le_i - VE_i * Le_r;
G21 = VE_r;
G22 = VE_i;

H1 = real(V_H*E);
H2 = imag(V_H*E);
```

```
G = [G11 G12 ; G21 G22];
H = [H1 H2];

K_hat = [x(1) x(2) 0 0; x(2) x(1) x(3) 0; 0 x(3) x(1) x(4) ; 0 0 x(4) x(5)];
C_hat = [x(8) 0 0 0; 0 x(8) 0 0; 0 0 x(8) 0; 0 0 0 x(9)];

R = [C_hat K_hat];

% Define the cost function.
f = trace((R * G - H)' * (R * G - H));
```

B7 – Altered Constrained Eigenstructure Assignment

```
% Constrained Eigenstructure Assignment function script

% Written by Paul E Paquet
% July 6, 2009

%Structural Dynamics, Control, and Identification
%Department of Civil and Environmental Engineering
%University of Waterloo, Waterloo, Ontario, Canada

% Define the function that produces the optimization cost function f.
function [f,dJ] = InTwoStorey(x,G,H)

% Calling the properties for the 2 storey model (included are the new
% eigenvalues and eigenvectors
ModProp

% Find the needed values for 2 measured eigenvalues.
for i = 1 : 2
    Gamma(:,:,i) = [(M * (L_mea(i) ^ 2) + C * L_mea(i) + K) eye(4)];
    Vi(:,:,i) = null(Gamma(:,:,i));
    Vik(:,:,i) = Vi(1:4,:,i);
    V_hat(:,:,i) = Vi(5:8,:,i);
    e(:,:,i) = (inv(Vik(:,:,i))) * V_mea(1:4,i);
end

% Ensuring that the matrices have the proper dimensions
E = zeros(8,2);
V(:,:) = zeros(4,8);
V_H(:,:) = zeros(4,8);
V(:,:) = [Vik(:,:,1) Vik(:,:,2)];
V_H(:,:) = [V_hat(:,:,1) V_hat(:,:,2)];

for i = 1 : 2
    E(1:4,i) = e(:,:,i);
end

% Setting up the matrices G and H by defining and calculating all of the
% values used.
Le = diag(L_mea).^2;
Le_r = real(Le);
Le_i = imag(Le);

VE_r(:,:) = real(V*E);
VE_i(:,:) = imag(V*E);

G11 = VE_r * Le_r - VE_i * Le_i;
G12 = VE_r * Le_i - VE_i * Le_r;
G21 = VE_r;
G22 = VE_i;

H1 = real(V_H*E);
H2 = imag(V_H*E);

G = [G11 G12 ; G21 G22];
```

```
H = [H1 H2];

K_hat = [x(1) x(2) 0 0; x(2) x(1) x(2) 0; 0 x(2) x(1) x(2) ; 0 0 x(2) x(3)];
M_hat = [[x(8) 0 ; 0 x(9)] zeros(2); zeros(2) [x(10) 0 ; 0 x(11)]];

R = [M_hat K_hat];

S = (R * G - H)';
S1 = R * G - H;

% Define the cost function.
f = trace(S * S1);
```

C1 – Full Scale Model Using Direct Stiffness

Modes	Analytical Eigenvalues	Measured Eigenvalues	Updated System Eigenvalues	Percent Error (%)	MAC
1	17.01	17.55	17.55	0.00	1.00
2	33.33	34.08	34.08	0.00	1.00
3	77.79	78.13	78.13	0.00	1.00
4	294.93	295.49	295.49	0.00	1.00
5	358.36	359.33	359.33	0.00	1.00
6	446.49	447.02	447.02	0.00	1.00
7	832.67	833.97	833.97	0.00	1.00
8	1460.90	1462.18	1462.18	0.00	1.00
9	1779.90	1778.80	1778.80	0.00	1.00
10	3205.60	3205.64	3205.64	0.00	1.00
11	4018.80	4021.76	4021.76	0.00	1.00
12	4951.80	4952.40	4952.40	0.00	1.00
13	5705.00	5704.82	5704.82	0.00	1.00
14	5705.00	5713.55	5713.55	0.00	1.00
15	7551.10	7559.01	7559.01	0.00	1.00
16	8200.40	8197.37	8197.37	0.00	1.00
17	9759.80	9756.42	9756.42	0.00	1.00
18	11179.00	11194.39	11194.39	0.00	1.00
19	14236.00	11194.39	14247.25	0.00	1.00
20	17557.00	14247.25	17558.51	0.00	1.00
21	19593.00	17558.51	19599.48	0.00	1.00
22	20105.00	19599.48	20111.36	0.00	1.00
23	24386.00	20111.36	24392.65	0.00	1.00
24	25306.00	24392.65	25316.15	0.00	1.00
25	29495.00	25316.15	29496.48	0.00	1.00
26	37735.00	29496.48	37748.64	0.00	1.00
27	39112.00	37748.64	39118.27	0.00	1.00
28	40593.00	39118.27	40595.35	0.00	1.00
29	50457.00	40595.35	50459.34	0.00	1.00

30	52876.00	50459.34	52866.65	0.00	1.00
31	57374.00	52866.65	57376.08	0.00	1.00
32	63227.00	57376.08	63235.03	0.00	1.00
33	74809.00	63235.03	74832.75	0.00	1.00
34	85964.00	74832.75	85967.54	0.00	1.00
35	88226.00	85967.54	88237.17	0.00	1.00
36	110790.00	88237.17	110802.37	0.00	1.00
37	119930.00	110802.37	120001.52	0.00	1.00
38	128270.00	120001.52	128280.66	0.00	1.00
39	139110.00	128280.66	139116.52	0.00	1.00
40	150110.00	139116.52	150221.52	0.00	1.00
41	172580.00	150221.52	172666.51	0.00	1.00
42	185060.00	172666.51	185064.01	0.00	1.00

C2 – Full Scale Model Using Direct Raw Measurement

Modes	Analytical Eigenvalues	Measured Eigenvalues	Updated System Eigenvalues	Percent Error (%)	MAC
1	17.01	17.55	17.55	0.00	1.00
2	33.33	34.08	34.08	0.00	1.00
3	77.79	78.13	78.13	0.00	1.00
4	294.93	295.49	295.49	0.00	1.00
5	358.36	359.33	359.33	0.00	1.00
6	446.49	447.02	447.02	0.00	1.00
7	832.67	833.97	833.97	0.00	1.00
8	1460.90	1462.18	1462.18	0.00	1.00
9	1779.90	1778.80	1778.80	0.00	1.00
10	3205.60	3205.64	3205.64	0.00	1.00
11	4018.80	4021.76	4021.76	0.00	1.00
12	4951.80	4952.40	4952.40	0.00	1.00
13	5705.00	5704.82	5704.82	0.00	1.00
14	5705.00	5713.55	5713.55	0.00	1.00
15	7551.10	7559.01	7559.01	0.00	1.00
16	8200.40	8197.37	8197.37	0.00	1.00
17	9759.80	9756.42	9756.42	0.00	1.00
18	11179.00	11194.39	11194.39	0.00	1.00
19	14236.00	11194.39	14247.25	0.00	1.00
20	17557.00	14247.25	17558.51	0.00	1.00
21	19593.00	17558.51	19599.48	0.00	1.00
22	20105.00	19599.48	20111.36	0.00	1.00
23	24386.00	20111.36	24392.65	0.00	1.00
24	25306.00	24392.65	25316.15	0.00	1.00
25	29495.00	25316.15	29496.48	0.00	1.00
26	37735.00	29496.48	37748.64	0.00	1.00
27	39112.00	37748.64	39118.27	0.00	1.00
28	40593.00	39118.27	40595.35	0.00	1.00
29	50457.00	40595.35	50459.34	0.00	1.00

30	52876.00	50459.34	52866.65	0.00	1.00
31	57374.00	52866.65	57376.08	0.00	1.00
32	63227.00	57376.08	63235.03	0.00	1.00
33	74809.00	63235.03	74832.75	0.00	1.00
34	85964.00	74832.75	85967.54	0.00	1.00
35	88226.00	85967.54	88237.17	0.00	1.00
36	110790.00	88237.17	110802.37	0.00	1.00
37	119930.00	110802.37	120001.52	0.00	1.00
38	128270.00	120001.52	128280.66	0.00	1.00
39	139110.00	128280.66	139116.52	0.00	1.00
40	150110.00	139116.52	150221.51	0.00	1.00
41	172580.00	150221.52	172666.51	0.00	1.00
42	185060.00	172666.51	185063.99	0.00	1.00

C3 – Full Scale Model Using Penalty Function

Modes	Analytical Eigenvalues	Measured Eigenvalues	Updated System Eigenvalues	Percent Error (%)	MAC
1	17.01	17.55	17.66	0.63	1.00
2	33.33	34.08	33.91	0.50	1.00
3	77.79	78.13	78.03	0.12	1.00
4	294.93	295.49	295.58	0.03	1.00
5	358.36	359.33	359.93	0.17	1.00
6	446.49	447.02	446.79	0.05	1.00
7	832.67	833.97	834.17	0.02	1.00
8	1460.90	1462.18	1461.88	0.02	1.00
9	1779.90	1778.80	1777.50	0.07	1.00
10	3205.60	3205.64	3204.96	0.02	1.00
11	4018.80	4021.76	4022.22	0.01	1.00
12	4951.80	4952.40	4951.86	0.01	1.00
13	5705.00	5704.82	5704.19	0.01	1.00
14	5705.00	5713.55	5711.11	0.04	1.00
15	7551.10	7559.01	7556.73	0.03	1.00
16	8200.40	8197.37	8196.68	0.01	1.00
17	9759.80	9756.42	9755.75	0.01	1.00
18	11179.00	11194.39	11193.71	0.01	1.00
19	14236.00	11194.39	14246.69	0.00	1.00
20	17557.00	14247.25	17559.80	0.01	1.00
21	19593.00	17558.51	19592.75	0.03	1.00
22	20105.00	19599.48	20112.49	0.01	1.00
23	24386.00	20111.36	24392.39	0.00	1.00
24	25306.00	24392.65	25323.46	0.03	1.00
25	29495.00	25316.15	29496.45	0.00	1.00
26	37735.00	29496.48	37750.84	0.01	1.00
27	39112.00	37748.64	39121.20	0.01	1.00
28	40593.00	39118.27	40605.81	0.03	1.00
29	50457.00	40595.35	50463.06	0.01	1.00

30	52876.00	50459.34	52888.74	0.04	1.00
31	57374.00	52866.65	57380.49	0.01	1.00
32	63227.00	57376.08	63227.28	0.01	1.00
33	74809.00	63235.03	74885.09	0.07	1.00
34	85964.00	74832.75	85986.67	0.02	1.00
35	88226.00	85967.54	88249.87	0.01	1.00
36	110790.00	88237.17	110781.18	0.02	1.00
37	119930.00	110802.37	120036.70	0.03	1.00
38	128270.00	120001.52	128166.87	0.09	1.00
39	139110.00	128280.66	139127.11	0.01	1.00
40	150110.00	139116.52	150268.07	0.03	1.00
41	172580.00	150221.52	172723.50	0.03	1.00
42	185060.00	172666.51	185104.60	0.02	1.00

C4 – Full Scale Model Using Quadratic Pencil

Modes	Analytical Eigenvalues	Measured Eigenvalues	Updated System Eigenvalues	Percent Error (%)	MAC
1	17.01	17.55	17.55	0.00	1.00
2	33.33	34.08	34.10	0.03	1.00
3	77.79	78.13	78.15	0.02	1.00
4	294.93	295.49	295.53	0.01	1.00
5	358.36	359.33	359.32	0.00	1.00
6	446.49	447.02	446.98	0.01	1.00
7	832.67	833.97	834.03	0.01	1.00
8	1460.90	1462.18	1462.28	0.01	1.00
9	1779.90	1778.80	1778.70	0.01	1.00
10	3205.60	3205.64	3205.81	0.01	1.00
11	4018.80	4021.76	4021.86	0.00	1.00
12	4951.80	4952.40	4953.09	0.01	1.00
13	5705.00	5704.82	5707.06	0.04	1.00
14	5705.00	5713.55	5710.18	0.06	1.00
15	7551.10	7559.01	7558.89	0.00	1.00
16	8200.40	8197.37	8197.05	0.00	1.00
17	9759.80	9756.42	9756.22	0.00	1.00
18	11179.00	11194.39	11193.71	0.01	1.00
19	14236.00	11194.39	14247.01	0.00	1.00
20	17557.00	14247.25	17558.53	0.00	1.00
21	19593.00	17558.51	19598.62	0.00	1.00
22	20105.00	19599.48	20111.06	0.00	1.00
23	24386.00	20111.36	24393.69	0.00	1.00
24	25306.00	24392.65	25314.61	0.01	1.00
25	29495.00	25316.15	29496.24	0.00	1.00
26	37735.00	29496.48	37748.36	0.00	1.00
27	39112.00	37748.64	39116.21	0.01	1.00
28	40593.00	39118.27	40595.32	0.00	1.00
29	50457.00	40595.35	50459.39	0.00	1.00

30	52876.00	50459.34	52865.85	0.00	1.00
31	57374.00	52866.65	57376.08	0.00	1.00
32	63227.00	57376.08	63234.94	0.00	1.00
33	74809.00	63235.03	74832.74	0.00	1.00
34	85964.00	74832.75	85980.10	0.01	1.00
35	88226.00	85967.54	88223.45	0.02	1.00
36	110790.00	88237.17	110802.62	0.00	1.00
37	119930.00	110802.37	120000.10	0.00	1.00
38	128270.00	120001.52	128280.49	0.00	1.00
39	139110.00	128280.66	139116.20	0.00	1.00
40	150110.00	139116.52	150216.65	0.00	1.00
41	172580.00	150221.52	172658.46	0.00	1.00
42	185060.00	172666.51	185059.70	0.00	1.00

C5 – Full Scale Model Using State Space

Modes	Analytical Eigenvalues	Measured Eigenvalues	Updated System Eigenvalues	Percent Error (%)
1	17.012	16.997	16.997	0.00
2	33.326	33.337	33.337	0.00
3	358.31	358.248	358.248	0.00
4	446.67	446.7898	446.7898	0.00
5	832.89	832.930	832.930	0.00
6	1463.00	1462.992	1462.992	0.00

C6 – Full Scale Model Using Constrained Eigenstructure Assignment

Modes	Analytical Eigenvalues	Measured Eigenvalues	Updated System Eigenvalues	Percent Error (%)	MAC
1	17.01	17.55	17.62	0.39	1.00
2	33.33	34.08	33.20	2.61	1.00
3	77.79	78.13	77.66	0.63	1.00
4	294.93	295.49	295.07	0.15	1.00
5	358.36	359.33	359.11	0.06	1.00
6	446.49	447.02	446.34	0.14	1.00
7	832.67	833.97	833.83	0.02	1.00
8	1460.90	1462.18	1460.84	0.10	1.00
9	1779.90	1778.80	1777.54	0.06	1.00
10	3205.60	3205.64	3204.74	0.03	1.00
11	4018.80	4021.76	4020.35	0.04	1.00
12	4951.80	4952.40	4952.08	0.02	1.00
13	5705.00	5704.82	5705.75	0.02	0.99
14	5705.00	5713.55	5708.93	0.02	0.77
15	7551.10	7559.01	7555.85	0.04	1.00
16	8200.40	8197.37	8195.68	0.02	1.00
17	9759.80	9756.42	9754.97	0.01	1.00
18	11179.00	11194.39	11192.63	0.01	1.00
19	14236.00	11194.39	14246.45	0.00	1.00
20	17557.00	14247.25	17557.12	0.01	1.00
21	19593.00	17558.51	19596.46	0.01	1.00
22	20105.00	19599.48	20110.05	0.00	1.00
23	24386.00	20111.36	24392.65	0.00	1.00
24	25306.00	24392.65	25309.21	0.02	1.00
25	29495.00	25316.15	29494.42	0.01	1.00
26	37735.00	29496.48	37747.18	0.00	1.00
27	39112.00	37748.64	39113.75	0.01	1.00
28	40593.00	39118.27	40593.45	0.00	1.00
29	50457.00	40595.35	50457.51	0.00	1.00

30	52876.00	50459.34	52860.13	0.01	1.00
31	57374.00	52866.65	57374.23	0.00	1.00
32	63227.00	57376.08	63232.49	0.00	1.00
33	74809.00	63235.03	74829.49	0.00	1.00
34	85964.00	74832.75	85974.12	0.01	1.00
35	88226.00	85967.54	88221.07	0.00	1.00
36	110790.00	88237.17	110800.20	0.00	1.00
37	119930.00	110802.37	119994.07	0.01	1.00
38	128270.00	120001.52	128278.15	0.00	1.00
39	139110.00	128280.66	139113.93	0.00	1.00
40	150110.00	139116.52	150210.55	0.00	1.00
41	172580.00	150221.52	172652.39	0.00	1.00
42	185060.00	172666.51	185053.83	0.00	1.00

C7 – Full Scale Model Using Altered Constrained Eigenstructure Assignment

Modes	Analytical Eigenvalues	Measured Eigenvalues	Updated System Eigenvalues	Percent Error (%)	MAC
1	17.01	17.55	17.01	3.07	1.00
2	33.33	34.08	33.33	2.22	1.00
3	77.79	78.13	77.66	0.63	1.00
4	294.93	295.49	295.07	0.15	1.00
5	358.36	359.33	358.34	0.27	1.00
6	446.49	447.02	446.38	0.14	1.00
7	832.67	833.97	833.08	0.11	1.00
8	1460.90	1462.18	1460.93	0.09	1.00
9	1779.90	1778.80	1777.30	0.08	1.00
10	3205.60	3205.64	3204.76	0.03	1.00
11	4018.80	4021.76	4019.52	0.06	1.00
12	4951.80	4952.40	4952.13	0.02	1.00
13	5705.00	5704.82	5705.83	0.02	1.00
14	5705.00	5713.55	5708.24	0.03	1.00
15	7551.10	7559.01	7555.40	0.05	1.00
16	8200.40	8197.37	8195.69	0.02	1.00
17	9759.80	9756.42	9754.49	0.02	1.00
18	11179.00	11194.39	11192.61	0.01	1.00
19	14236.00	11194.39	14246.12	0.01	1.00
20	17557.00	14247.25	17557.01	0.01	1.00
21	19593.00	17558.51	19595.19	0.02	1.00
22	20105.00	19599.48	20109.81	0.01	1.00
23	24386.00	20111.36	24392.72	0.00	1.00
24	25306.00	24392.65	25309.18	0.02	1.00
25	29495.00	25316.15	29494.48	0.01	1.00
26	37735.00	29496.48	37747.31	0.00	1.00
27	39112.00	37748.64	39112.17	0.01	1.00
28	40593.00	39118.27	40593.50	0.00	1.00
29	50457.00	40595.35	50457.56	0.00	1.00

30	52876.00	50459.34	52860.14	0.01	1.00
31	57374.00	52866.65	57374.27	0.00	1.00
32	63227.00	57376.08	63230.69	0.01	1.00
33	74809.00	63235.03	74829.79	0.00	1.00
34	85964.00	74832.75	85974.18	0.01	1.00
35	88226.00	85967.54	88219.11	0.00	1.00
36	110790.00	88237.17	110798.23	0.00	1.00
37	119930.00	110802.37	119994.06	0.01	1.00
38	128270.00	120001.52	128276.16	0.00	1.00
39	139110.00	128280.66	139112.04	0.00	1.00
40	150110.00	139116.52	150210.55	0.00	1.00
41	172580.00	150221.52	172652.39	0.00	1.00
42	185060.00	172666.51	185053.83	0.00	1.00